1998

The application of virtual reality as a user interface paradigm for telecommunication network management

Mohsen Kahani
University of Wollongong

The Application of Virtual Reality as a User Interface Paradigm for Telecommunication Network Management

A thesis submitted in fulfilment of the requirements for the award of the degree of

Doctor of Philosophy

from

University of Wollongong

by

Mohsen Kahani

B.E. (1990) University of Tehran,

M.Eng.Stud. (1993) University of Wollongong

School of Electrical, Computer and Telecommunication Engineering

(1998)

بسم الله الرحمن الرحيم

In the name of God, the merciful and compassionate

Declaration

This is to certify that the work reported in this thesis was done by the author, unless specified otherwise, and that no part of it has been submitted in a thesis to any other university or similar institution.

Mohsen Kahani

Acknowledgments

Praise to God, whose blessings helped me come to this stage.

A few people helped me in different stages of thesis preparation. First, I would like to thank my Supervisor, Dr H.W. Peter Beadle, for his invaluable advice and support throughout the project. I would like to extend my appreciation to Professor Gary J. Anido, my second Supervisor.

I would like to acknowledge the Ministry of Culture and Higher Education (MCHE) of the Islamic Republic of Iran for awarding me a generous scholarship, consisting of tuition fees and stipend. In addition, the assistance of The Institute for Telecommunications Research (TITR) at the University of Wollongong for the provision of funding, for purchasing equipment and conference attendance, is hereby acknowledged.

I express my gratitude to the staff of the School of Electrical, Computer and Telecommunications Engineering and my fellow postgraduate colleagues for provision of an enjoyable environment, in which I could pursue the research. In addition, many thanks to those who participated in the evaluation phase of the WWW-based user interface. I also would like to acknowledge Ms. Maree Fryer for proof reading the thesis.

Last, by not least, I would like to express my greatest appreciation to my wife Homa and my children Fatemeh and Mohammad. Without their encouragement, understanding and support this thesis simply could never have come into existence.

This humble work is dedicated to the **GREAT NATION OF IRAN.**

Abstract

Networks are becoming increasingly complex and automated. This is due in part to their increased bandwidth, the need for automated protection switching, the deployment of virtual networks based on B-ISDN and ATM technology and the increase in traffic complexity typified by multimedia communications. In turn, network management is becoming more complex and more mission-critical to a larger number of organisations. In the past, this has led to the development of integrated network management systems using Windows Icons Mouse Pointer (WIMP) based user interfaces. However, it is believed that these two-dimensional user interfaces are no longer suitable for management of large and high-speed broadband networks.

This thesis investigates the application of Virtual Reality (VR) user interfaces for management of modern telecommunication networks. It tries to exploit the extra display dimensions and high level of interaction of VR systems to provide an environment, in which managers can navigate and manipulate network elements more easily, reducing the risk of human errors. Moreover, the thesis is aimed at determining the usability of 3D user interfaces for network management and comparing them with traditional network management systems.

In this regard, initially, an immersive user interface has been designed and implemented. In this system, users use head mounted displays (HMDs), VR glove and other 3D input and output devices to navigate through the network hierarchy and manipulate the

network elements. Object-oriented design techniques were used to create a modular and configurable system.

The empirical and qualitative evaluation of this system revealed that the required technology for using immersive VR as a user interface paradigm is not yet available. This fact led to the development of a desktop flat screen 3D VR user interface based on WWW-technologies. This permits the use of tools and facilities provided by the Internet to create a distributed collaborative multiuser interface.

A protocol is required for the exchange of collaboration data among the users. As the existing protocols, such as DIS, were inadequate, a protocol and its Protocol Data Units (PDUs) were defined. Realising the similarities among user interfaces of different applications, the protocol was generalised to become a Generic Protocol for Multiuser Interfaces (GPMI).

Using the GPMI, a Web-based multiuser interface for network management was designed and implemented. The WWW-based Network Management System (WNMS) uses the facilities of a three-dimensional desktop VR to provide an integrated universal interface, in which users collaboratively manage networks more efficiently. WWW technologies, such as VRML, HTML, CGI scripts, Java and JavaScript, were utilised to implement this system.

Apart from empirical evaluation methods, more rigorous techniques were used to compare WNMS with one of the traditional network management systems, showing the merits and pitfalls of each system. The experiments and subjective measurements show that the users significantly preferred the WNMS to the traditional user interface. In addition, the measurement of the taskload using the NASA TaskLoad Index (TLX) revealed that the amount of cognitive workload in WNMS is significantly less than that of the traditional user interface.

Table of Contents

Chapter 3 – Immersive Virtual Reality User Interface

Chapter 6 – Quantitative Evaluation and Comparison

Chapter 7 – Conclusion and Further Research

References

Appendices

List of Abbreviations

ATM	Asynchronous Transfer Mode
BISDN	Broadband Integrated Services Digital Network
CCITT	Former name of the ITU-T Standard Organisation
CM	Collaboration Manager
CP	Community Place
DIVE	Distributed Interactive Virtual Environment
DVR	Distributed Virtual Reality
DIS	Distributed Interactive Simulation
GPMI	Generic Protocol for Multiuser Interfaces
HTML	HyperText Markup Language
HTTP	HyperText Trasfer Protocol
ISDN	Integrated Services Digital Network
ISO	International Standard Organisation
JMAPI	Java Management API
MASSIVE	Model, Architecture and System for Spatial action in Virtual Environment
MUI	MultiUser Interface
NE	Network Element
NMS	Network Management System
NPSNET	Naval Postgraduate School NETwork
NVR	Networked Virtual Reality
OODB	Object-Oriented DataBase
OSI	Open System Interconnection
QoS	Quality of Service
SIMNET	SIMulated NETworking
SNMP	Simple Network management Protocol
TCP/IP	Transport Control Protocol/Internet Protocol

TMN	Telecommunication Management Network
VC	Virtual Circuit
VP	Virtual Path
VR	Virtual Reality
VRML	Virtual Reality Modelling Language
WBEM	Web-Based Enterprise Management
WNMS	WWW-based Network Management System
WWW	World Wide Web

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

Performance improvements in network management systems are currently limited, in part, by network management user interfaces. This thesis proposes and quantitatively demonstrates that Virtual reality (VR) can provide a better network management user interface than window-based systems.

The experimental investigation consists of the development of two types of 3D user interfaces for network management, and a qualitative and quantitative comparison of them with conventional network management user interfaces to show the superiority of virtual reality ones. An immersive virtual reality user interface was constructed using a head-mounted display, virtual glove and other 3D input devices. Qualitative evaluation of this system indicated that the current technology was not yet suitable for the application, but that three-dimensional display showed promise. A 3D WWW-based desktop VR system was then investigated using the Web browsers for the user interface.

Collaboration and support for distributed management were added to this system to allow several network operators to manage big network collaboratively and from geographically dispersed locations. To support the collaboration issues of this user interface a generic protocol for multiuser interfaces was developed. Finally, the system was quantitatively compared with traditional two-dimensional window based user interfaces showing that the VR system was superior.

1.2 Background and Motivation

Managing telecommunication networks is one of the important issues of future networks. In these networks, each network device generates a significant quantity of information about its status. For instance, the status information generated by each digital exchange on average is more than one megabyte per minute [Rea, 1993]. This information can be employed to monitor the performance of the network and maintain a desired service level. The emergence of new services, such as voice and video, will increase the complexity involved in monitoring and controlling the network. This level of complexity urges the manager to move from simple network management systems to more effective ones.

Moreover, the trend in telecommunications is towards building more intelligent and automatic network elements, so that if a problem occurs, the elements can quickly perform actions to rectify it [Rea, 1993]. In addition, using expert systems in network management is quite popular today. These systems try to detect problems, and automatically solve some of them. This avoids overwhelming the manager with trivial alarms. However, most decision-making processes are still human-based. As the networks grow, the importance of the decision made by the operators increases, so that a wrong decision can have catastrophic effects on the network.

To reduce the risk of human errors, user interface techniques have been improved. Textual, command line function-oriented user interfaces have been replaced by WIMP (Window, Icon, Mouse and Pointer) object-oriented systems to reduce the network operators' cognitive load. These new interfaces use direct manipulation techniques [Shneidermann, 1983]. Consequently, they provide good visualisation of the system, which improves the understanding of data. In WIMP environments, each Network Element (NE) is represented by an icon within a window. By clicking on the icon, further details of the device appear in another window. Messages and the actions of alarm correlation and diagnosis expert systems are presented through text information boxes. Figure 1 shows a typical view of such user interfaces.

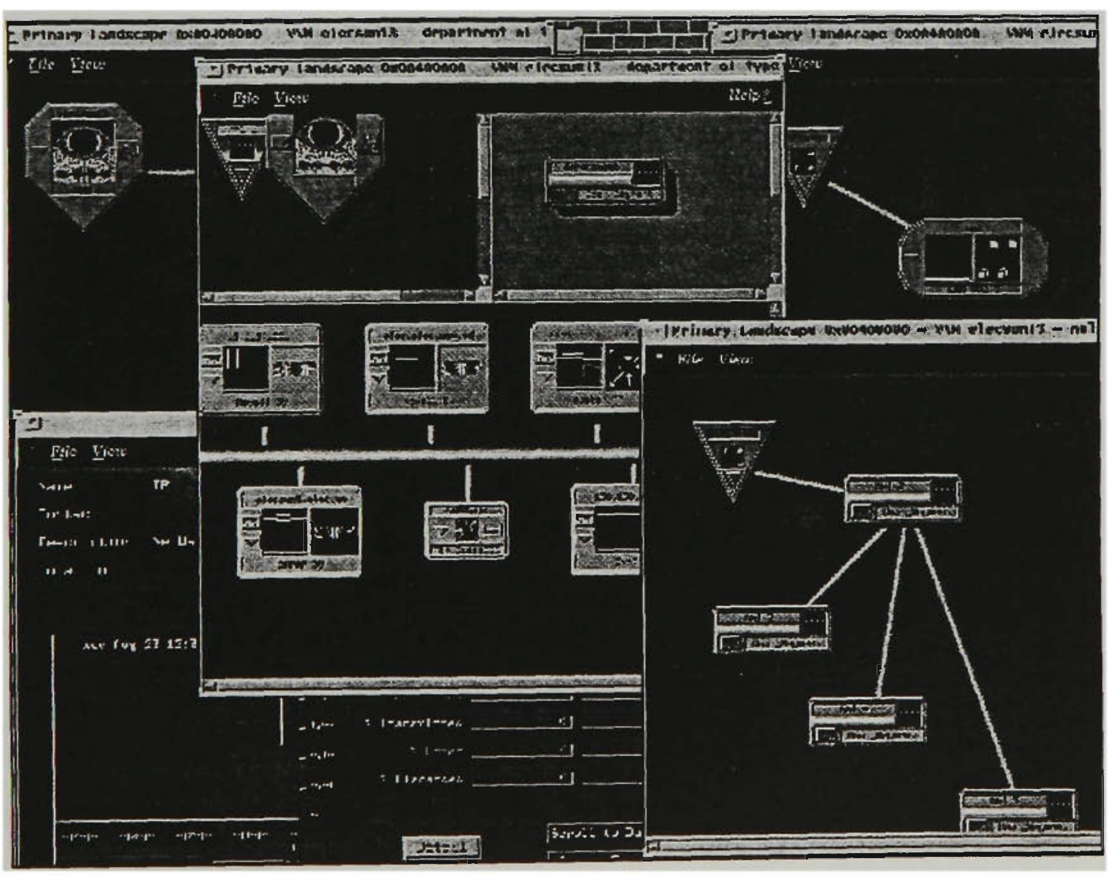

Figure 1.1- A typical view of WIMP user interfaces (Spectrum)

Two-dimensional user interfaces appear to be inadequate for visualising and manipulating big databases and systems [Lazar, 1992]. Managing future high-speed networks based on Broadband Integrated Services Digital Network (BISDN) requires quick understanding and navigating through large datasets and spatial networking structure, as well as higher levels of automation. Therefore, it will be difficult, if not impossible, for traditional user interfaces to cope with such levels of complexity. In general, the Current 2D management workstations are limited in three areas [Crutcher, 1993]:

- Monitoring bandwidth: The throughput of management information becomes lower, because of the complexity of management protocols and low bandwidth of management connections.

- Semantic level of display: Although graphic display is an advance over textual ones, it still represents a barrier between the behaviour of the network and what is perceived by the user.

- Level of interaction: Because of the above limitations and lack of appropriate input devices, the interaction of the user with the system is low. Hence, the user is typically passive and observes the network rather than actively directs its operation.

Some of these limitations can be well addressed in a 3D virtual reality environment consisting of high performance 3D image processors, 3D displays and interaction devices, and with the high bandwidth capabilities of gigabit networks. These kinds of interfaces, firstly, offer an external spatial dimension, which improves the semantic level of display. The additional bandwidth capabilities allow quick fault discovery and removal. Finally, as the user's input is explicit in terms of the objects to be manipulated, a higher level of immersion and interaction can be provided [Crutcher, 1993].

The idea of VR interfaces is also driven by the increased level of integration and complexity in emerging networks, and the need for visualising the repercussions of each management decision on the overall performance of the network, before the decision is put into practice. The virtual network capability of these networks, which means their bandwidth allocation and configuration can be dynamically reconfigured is also another factor.

One of the major problems of using most traditional user interfaces is the long training period of operators. In a well-designed VR environment, the user's interaction with the system is as natural as possible. Therefore, the need to teach operators how to use the system is significantly reduced. That is, if operators learn the basic principals of the interface, they can easily and quickly decide how to do the task in more complicated situations.

1.2.1 Psychological considerations

It is also desirable to consider the user interface from a psychological point of view. Figure 1.2 illustrates several stages of the user's interaction with the computer. In the execution of a goal, the first step is intention. Intention is the decision to act to achieve the goal. The intention is translated into a sequence of actions to perform the task. These actions are executed using the physical mechanism provided by the interface [Norman 86].

Consider the reading of email as the goal. This goal is translated into the intention of running the email reader program. Action specification for this intention is finding the appropriate icon and double clicking on that. Finally, one might use a mouse to execute those actions.

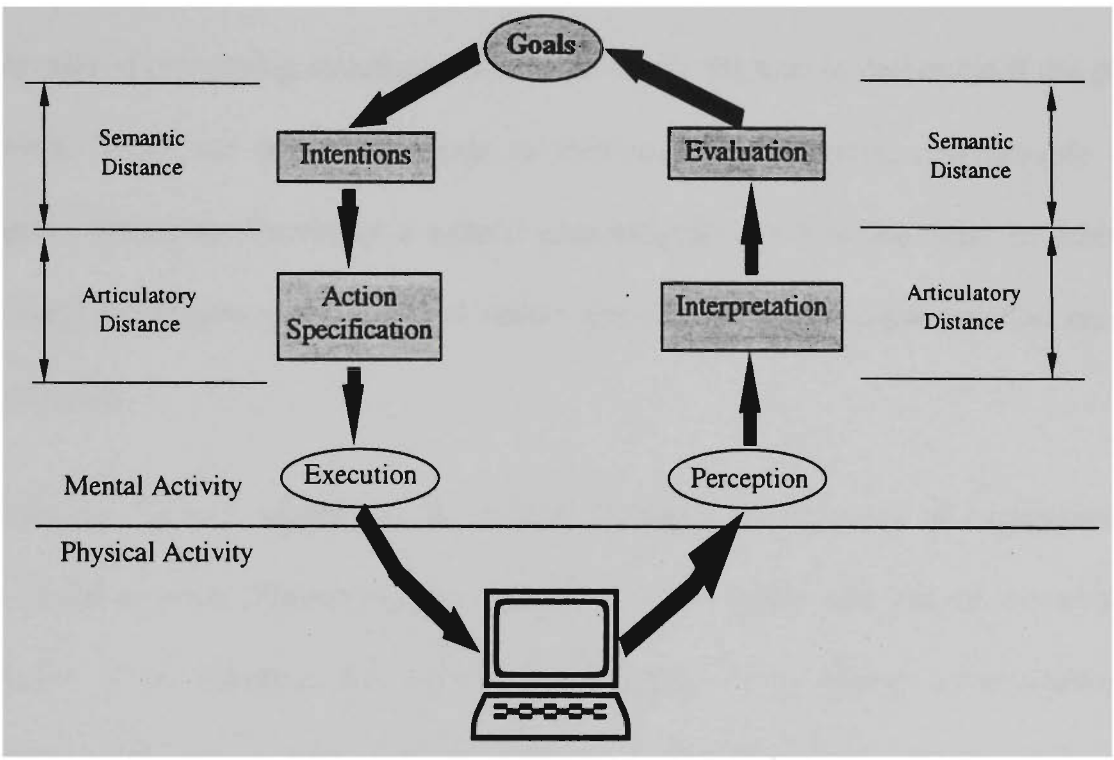

Figure 1.2- Several stages of human computer interaction [Norman 86].

After executing the actions, the outcome should be evaluated. The result is shown on the display interface of the system, and is interpreted by the user. Finally, the interpretation of the state of the system and the original goal are compared to evaluate whether the goal has been achieved or further actions are required. For the above example, after double clicking by mouse, a window will appear on the screen. This window is interpreted as the email reader output, and is compared with the original goal. In case another program has been mistakenly clicked, the evaluation fails, and the cycle might be repeated.

It is also possible to express the distance between the goal and the physical system in terms of semantic distance and articulatory distance [Hutchine, 1986]. Semantic distance on the input side represents the distance between the goal of the user and the meaning of the goal in the interface language. On the output side, however, it refers to the amount of processing structure that is required for the user to determine if the goal is achieved. There are several methods to shorten these distances, and provide more semantic directness. Providing a natural user interface, so that the input matches the real-world interaction and the output shows some semantic concepts directly, shortens this distance.

Articulatory distance represents the relation between the meaning of expression and their physical form. Shortening this distance on the input side can be achieved by provision of an interface that permits specification of an action by mimicking it. Similarly, on the output side, providing symbols that directly represent the variables that the user wants to control, provides more articulatory directness. Note that an articulatory direct interaction can couple the interaction between actions and their meanings so directly that relationship between intentions and actions and between actions and output seems straightforward and obvious [Hutchine, 1986].

Some psychologists argue that an interface by nature is a form of artistic imitation, a mimesis, which tries to mimic real life situations [Laurel, 1986]. It is only through providing engagement that an interface can help the user meet the goals. *First-personness*, rather than second or third personness, can provide that level of engagement. That is, direct engagement occurs if users experience direct interaction with the objects in the domain, similar to first-person games, such as Doom and Quake. In those environments, there is a feeling of involvement directly with a world of objects rather than of communicating with an intermediary [Hutchine, 1986].

To provide a mimetic world, which emulates the real world, design principle should be formulated based on a set of minimum requirements. First, both execution and evaluation should be direct. Also, the input and output language have to be interreferential, in the sense that an input expression can make use of a previous output expression. The other principle is the responsiveness of the system. There should not be any delay between the execution and the result, except when the delays are appropriate for the knowledge domain itself. The last requirement is that the interface has to be unobtrusive, and neither interfering nor intruding. If the user feels the presence of the interface, the first-personness feeling tends to fades, and detracts from the directness of the engagement [Laurel, 1986].

1.2.2 Need for distribution and collaboration

The Broadband Integrated Services Digital Network (BISDN) based on Asynchronous Transfer Mode (ATM) technology offers huge bandwidth capabilities, making the emergence of sophisticated multimedia applications possible. ATM networks include the concept of logical connectivity and virtual private network (VPN) [Kositpaiboon, 1993]. A virtual private network is a set of network resources, such as user-network interfaces (UNIs), and (semi) permanent virtual connections (PVC) that link the different sites of a customer together, as shown in Figure 1.3. However, this logical connectivity, although providing higher management flexibility than physical connectivity, increases the complexity of the network management task.

The virtual private network concept also implies that there are some dependencies between the operation of different networks, because they may share the same physical link. Looking at Figure 1.3 shows that Corporation X and Corporation Y are using the same physical public ATM network, which if it fails, affects both of them. Consequently, some kind of collaboration among network management systems of private networks with that of the carrier is required to effectively manage the network in real time. This provision has also been suggested by ATM Customer Network Management (CNM) specification [ATM Forum-CNM, 1996].

Managing ATM networks also requires a more decentralised approach. Several organisations, from the network provider to customer site administrators, may require hierarchical access to the network management information. That means that while distributing the management, a centralised and integrated view of the whole system should also be provided for efficient network management.

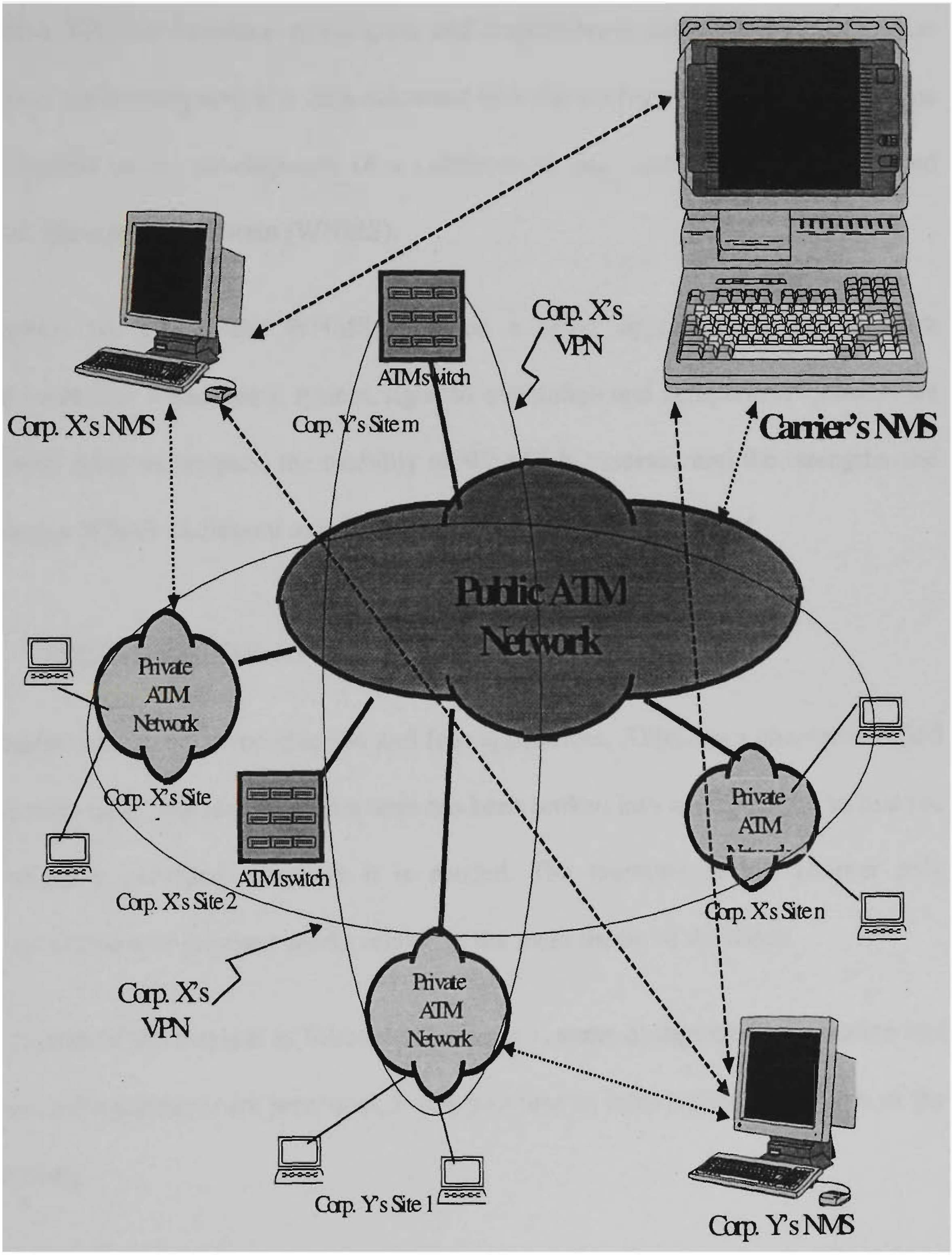

Figure 1.3- An example of virtual private networks

Considering these requirements, this thesis investigates the application of collaborative multiuser VR user interfaces for telecommunication network management. An immersive VR user interface is designed and implemented and several visualisation techniques are investigated. It is then discussed how the observation from this prototype system leaded to the development of a collaborative and distributed 3D Web-based Network Management System (WNMS).

To support the claim that WNMS provides a better approach for the network management than a traditional system, rigorous evaluation and comparison methods are used. With these techniques, the usability of WNMS is assessed and the strengths and weaknesses of both traditional user interfaces and WNMS are compared.

1.3 Organisation of the Thesis

This thesis consists of seven chapters and four appendices. Although a chapter is called 'literature review', the review of literature has been broken into several parts, so that the information is provided whenever it is needed. The literature review chapter only provides a review of previous works related to the main theme of the thesis.

The structure of the thesis is as follows. In Chapter 1, some background information and motivation for the thesis are presented. It also provides an introduction for the rest of the dissertation.

The literature review is given in Chapter 2. As the proposed system is a distributed and collaborative virtual reality system, distributed virtual reality systems are reviewed in this chapter. In addition, previous work on the provision of a three-dimensional virtual reality user interface for network management is discussed. Finally, a review of user interface evaluation techniques is presented.

The design and implementation issues of an immersive VR user interface for network management is discussed in Chapter 3. Firstly, the justification for using an immersive system is given. Then the design methodology, implementation issues, and observation from the system are presented. A qualitative evaluation of the system follows. Finally, the chapter explains the reasons for migrating to a WWW-based user interface.

To avoid forward referencing and repetition, Chapter 4 discusses the development of a protocol that addresses the collaboration issues of the VR user interface discussed in Chapter 5. In this chapter, the major issues regarding the collaboration between users in a 3D world is initially discussed. Then, the structure and PDUs of GPMI (Generic Protocol for Multiuser Interfaces) are described.

Chapter 5 utilises the protocol developed in Chapter 4 to design and implement a web-based 3D collaborative distributed user interface for network management (WNMS). The user interface is based on web-browsers enhanced with VRML (Virtual Reality Modeling Language), Java and JavaScript. Design implementation issues, as well as a qualitative evaluation of this system, and comparison with the immersive system are among the material discussed in this chapter.

Chapter 6 provides objective and subjective evaluation techniques to compare the proposed user interface with a traditional one, quantitatively. A set of guidelines is developed, against which both systems are compared.

The major findings of the thesis are summarised in Chapter 7. A number of suggestions for further research are also presented. In Appendix A, a detailed description of the classes used in the immersive system is presented. Specification of VR devices used in the immersive system is given in Appendix B. Appendix C contains the ASN.1 code of the GPMI PDUs, and finally, the user satisfaction questionnaire and the raw result are presented in Appendix D.

1.4. Contributions

This section lists the main contributions made in relation to this thesis, pointing to the chapters where they are discussed.

1. Design, implementation and qualitative evaluation of an object-oriented immersive 3D user interface for network management- Chapter 3.

2. Development of GPMI, a generic protocol to address the collaboration issues in multiuser interfaces- Chapter 4.

3. Proposal for a web-based approach (VRML) as a 3D graphical user interface for network management- Chapter 5.

4. Design, implementation and qualitative evaluation of a collaborative 3D multiuser interface for distributed management of networks using VRML, Java and other WWW technologies- Chapter 5.

5. An architecture for distributed management of multivendor heterogeneous networks- Chapter 5.

6. Development of a set of guidelines for quantitative evaluation of network management user interfaces- Chapter 6.

7. Experimental method to compare 3D WWW-based user interface with respect to one of the traditional 2D user interfaces for network management- Chapter 6.

1.5. Publications and Awards

1.5.1 Journal paper

- Kahani, M., Beadle, P., "Decentralised Approaches for Network Management", *ACM Computer Communication Review*, July 1997.

1.5.2 Full paper refereed conference papers

- Kahani, M., Beadle, P., "WWW-based 3D Distributed, Collaborative Virtual Environment for Telecommunication Network Management", *ATNAC'96 Conference Proceedings*, December 1996, Vol. 2, pp. 483-8.

- Kahani, M., Beadle, P., "Immersive and Non-immersive Virtual Reality Techniques Applied to Telecommunication Network Management", *Proceedings of IEEE/IFIP International Symposium on Integrated Network Management (IM97)*, May 1997, pp. 383-95.

1.5.3. Extended abstract refereed conference papers

- Kahani, M., Beadle, P., "Using Virtual Reality to Manage Broadband Telecommunication Networks", *ATNAC'95 Conference Proceedings*, Vol. 2, 1995, pp. 517-522.

- Kahani, M., Beadle, P., "Comparing Immersive and Non-Immersive Virtual Reality User Interfaces for Management of Telecommunication Networks", *Proceedings of IEEE International Conference on Telecommunication (ICT97)*, April 1997, pp. 1121-6.

- Kahani, M., Beadle, P., "Collaboration in Persistent Virtual Reality Multiuser Interfaces: Theory and Implementation", *Electronic Proceedings of Virtual Reality Universe (VRU97) Conference*, available at www.cyberedge.com/vru_paper/kahani.html, April 1997, CA, USA

1.5.4 Award

- IEEE Computer Society student travel grant for IEEE/IFIP International Symposium on Integrated Network Management (IM97), May 1997, San Diego, USA. This award was granted based on the scores that the referees gave to the paper.

Chapter 2

Literature Review

2.1 Introduction

The literature review consists of three sections. Initially, literature related to Distributed Virtual Reality (DVR) systems is reviewed. Then, the non-traditional user interfaces for network management systems are discussed. These user interfaces include some VR and non-VR systems developed to address problems encountered with current user interfaces for network management. These two sections, provide a background for the main theme of the thesis; a multiuser collaborative VR user interface for telecommunication network management.

In the third section, the evaluation methods and measuring techniques currently used for user interface evaluation are discussed. This section provides the required background for Chapter 6, in which the proposed user interface is compared with traditional user interfaces for network management.

2.2 Distributed Virtual Reality Systems

Distributed VR is one of the hottest issues related to virtual reality. The idea is to have a simulated environment, which runs simultaneously on several computers, connected together over a network. People using these computers, can interact in real-time, sharing the same virtual world [Roehl, 1995a].

Many systems have been developed to address the specific issues related to distributed VR systems. A brief discussion for some of the most important systems is presented in this section. While these systems have been chosen as a representative sample, it is by no means an inclusive list.

2.2.1 NSPNET

NPSNET is the Department of Computer Science, Naval Postgraduate School NETwork. NPSNET is a real-time simulator, which is used for military simulation, and several applications have been successfully implemented upon it [Locke, 1992]. Some related applications are discussed below.

The SIMulator NETworking (SIMNET) [Pope, 1989; Miller, 1995] project has been developed to provide technology for distributed simulation of large-scale combat over local and long haul networks. It focuses on simulating an armoured battlefield. The system is limited to three basic object types: Static (not moving), Simple (no articulated parts), and Tank (two-articulated parts, turret and gun).

The success of SIMNET led to the development of the Distributed Interactive Simulation (DIS) [IEEE-DIS, 1993] standard. DIS is the largest best-known standard for distributed VR. It uses a standard format, called Protocol Data Unit (PDU), to exchange information between simulation hosts. DIS uses a fully distributed model, in which no central server exists for event scheduling or conflict resolution. DIS is primarily designed for military simulations, and focuses on simulation of vehicle type objects in the virtual environment. To reduce the amount of traffic, DIS also employs dead reckoning technique. Using this technique, instead of sending each entity's position and orientation continuously, its velocity vector is transmitted. The receiver can use this information to calculate the entity's location at any later time.

Both SIMNET and DIS have been designed for small unit training and are not suitable for large-scale multiplayer VE's. They require enormous bandwidth and computation for large-scale simulation. As they use broadcasting for data transfer, the network segments should be bridged together.

To overcome the problem, a multicast-based system (NPSNET) has been proposed [Macedonia, 1995a; Macedonia, 1995b]. NPSNET uses hexagonal cells to divide the simulated area into several regions, called Areas Of Interest (AOI). Each user, depending on their locations in the virtual world, belongs to an area of interest. Each AOI is associated with a multicast group. Whenever an entity moves, its AOI changes, so it dynamically joins or leaves multicast groups. As the data is transferred via the network layer, the participants can be on geographically dispersed sites. However, the multicast protocol still has not been implemented on the whole network. Therefore, the MBONE protocol [Eriksson, 1994] is used to connect islands of multicast-enabled networks together.

The big advantage of the abovementioned systems, especially the last one, is their support for a high number of participants. Even in SIMNET, more than one hundred users could participate [Macedonia, 1995a]. This number is much higher than capabilities of other DVR systems. However, NPSNET has been built to support thousands of users.

This advantage is a result of compromising on other factors. In all of the systems mentioned above, there has been little consideration for semi-static environmental objects, called terrain. Therefore, as discussed in more detail in Chapter 4, the DIS protocol and its developments are not appropriate for systems where objects are important, such as user interface systems. Some researchers have tried to address several of these issues in the context of DIS by adding a number of PDUs [Mastgalio, 1985]. However, their success has yet to be seen.

Another issue in these systems is that all participants should have full replication of the database before they enter the simulation. This condition cannot be fulfilled in systems where the database is continuously evolving.

2.2.2 DIVE

Distributed Interactive Virtual Environment (DIVE) is a heterogeneous distributed VR system based on the Unix and Internet networking protocols, developed at the Swedish Institute of Computer Science (SICS) [Carlsson, 1993]. DIVE is especially tuned for multi-user applications, where several participants may interact, although they may be geographically dispersed.

The distribution model of DIVE is based on peer-to-peer communication, multicast protocol and coarse-grained data partitioning. Earlier versions of DIVE [Carlsson, 1993] used a full replication model with a reliable positive-acknowledgement multicast protocol. However, the latest version [Hagsand, 1996] uses partial replication and a multicast protocol based on a negative acknowledgement (nack) scheme. This model minimises end-to-end latency, and allows more participants.

To reduce the update data traffic, the latest version of DIVE uses the concepts of aura and an aura manager. The concept is similar to that of MASSIVE, and is explained in detail in the following subsections. However, it is different from the MASSIVE approach, as DIVE relies on a replicated database model, and uses the aura manager as a means to signal to clients which parts of the database need to be replicated [Hagsand, 1997].

DIVE uses explicit locking to prevent concurrent access to an object. It assumes that users "own" objects for a long time, and concurrent accesses rarely occur. Therefore, only the user that has the write token can modify the object. In case of a conflict, one user has to wait until it receives the token.

The earlier versions of DIVE had provided more restrict consistency, while the later one have relaxed some of the restrictions (by using nack instead of ack). Consequently, the latest versions are more scalable. Moreover, as DIVE uses a distributed algorithm for consistency control, it uses the concept of aura to limit the number of users involved in the decision making, and assumes users hold objects for a long time. These assumptions are not always true in multiuser interfaces. If desktop VR is used as the interface medium, users have access to very far , but visible, objects via pointing devices, such as a mouse. Therefore, the auras should be considered very large, which in fact makes them useless. In addition, users might not hold the objects for a long period.

2.2.3 AVIARY

Another distributed virtual reality system is AVIARY [Snowdon, 1993; Snowdon, 1994]. AVIARY is designed to be platform-independent, and provides an environment, which supports multiple concurrent virtual worlds, applications, users, and interaction among them. In this system, everything is an autonomous, concurrently executable object. The environment is divided into separate worlds. Similar to a GUI, several worlds can be concurrently active. Special objects, called 'portals', are provided to support movement between worlds.

The main components of AVIARY are its communications system, Virtual Environment Manager (VEM), Environment DataBase (EDB), object servers, input and output objects, user objects, applications and renderer. Objects communicate via the communications system, which allow them to send messages to each other, without knowing where the destination object resides. As the communications system also tracks creation and destruction of objects, the object can be created dynamically when needed.

VEM provides services that must be consistent throughout the system, such as generating unique IDs for objects. It also notifies holders of the shared objects of updates to the objects, and assigns integer identifiers to message selectors.

EDB provides some spatial management. It performs coarse collision detection, and informs objects involved in a collision. EDB is also used by the renderer to determine which artifacts are visible from a given viewpoint.

Object servers contain lightweight objects, which receive messages from other objects, handle execution of the relevant methods and provide simple scheduling facilities.

Although AVIARY was developed in 1992, it has attracted little attention in the literature. It may be because of the complexity of the model and architecture. Moreover, it has focused on issues related to the user and applications, and does not address the other semi-static objects usually present in a virtual world. Also, in the published papers all objects are shown in wire-frame, which is not suitable for immersive environments.

2.2.4 MASSIVE

Model, Architecture and System for Spatial action in Virtual Environment (MASSIVE) is a virtual reality teleconferencing system [Benford, 1993; Greenhalgh, 1995]. In this system, communication between participants is controlled by movement within a shared virtual space. MASSIVE allows groups of participants to communicate over different media using different interface equipment. It supports many simultaneous meetings, and the participants can be distributed over a wide area network.

The aim of MASSIVE is the provision of support for flexible management of communications in densely populated virtual spaces. The main components of the model are aura, awareness, focus, nimbus and adapters. Aura is a bounded sub-space around an object, which limits the presence of the object within a medium. This concept has been used to address the problem of limiting the number of connections between the participants. That is, the connection between two users is established only when their auras collide.

After a connection is established, the awareness, focus, and nimbus control the information passing across it. Nimbus represents the amount of control that the transmitter has over the information propagated to the other object, while focus represents the receiver's control. The awareness of an object, A, of object, B, in medium, M, is a combination of A's focus in M and B's nimbus in M. Adapters represent communication tools, boundaries, or other kinds of objects that provide a degree of extensibility. Adapters can alter other components of the model.

One of the main features of MASSIVE is its heterogeneity. That is, users can interact with each other over a combination of graphics, audio and text media. The user's terminal equipment varies from a sophisticated graphics workstation with audio and text clients to a VT-100 dumb terminal. To determine whether objects have any media in common, the system uses a dynamic brokering mechanism.

MASSIVE only addresses the issues related to the users, as it was intended. It introduces different degrees of presence for the objects, simulating the real-world situations. The concept of defining an aura for each user has received some attention, and it has been used in other systems, such as DIVE, though in a different context.

2.2.5 NVR

NVR (Networked Virtual Reality) is an API toolkit that can be used to create multiuser VR applications networked over TCP/IP networks [Berger, 1994]. It has been designed to fulfill the following goals: low cost, scalable, modular, flexible, easy to use, high performance, portable, and allow geographically dispersion, though it could not fulfill some of these goals, particularly scalability.

NVR employs a client-server communication architecture. All clients connect to a central station, the server, and communicate with each other through the server. It also utilises TCP/IP for transmission of data between client and server. It uses a technique called "thresholding", which means that clients should broadcast their changes every *n*th rendering frame. This prevents fast station from overwhelming the network, and slower machines.

The database in NVR is fully replicated, but the whole virtual world can be broken into several small worlds connected together via portals. Only one world has to be kept in the memory of the client's station at each time. Clients send their manipulation of the environment using the thresholding method, and they are stored in the server to keep newcomers updated.

NVR uses explicit locking to preserve consistency, and prevent concurrent modification of an object. Before accessing an object, the user should lock it, and wait until it gets a lock acceptance or rejection, before starting to modify the object. NVR does not provide any mechanism for dead lock avoidance, and it is the responsibility of the system developer to take that into consideration.

Practical experiments have shown that NVR does not scale up well [Berger, 1994]. Instead of using dead reckoning for update exchange, it uses thresholding, which causes increased traffic as the number of changes increases. In addition, it does not consider any technique for update filtering, such as AOI, so all information is sent to all participants regardless of whether they are interested or not.

2.2.6 Community Place

Community Place (CP) [Lea, 1996] has been developed by the Virtual Society project to support large-scale shared 3D spaces on the Internet using the Virtual Reality modeling Language (VRML) [Bell, 1996]. It uses its proprietary protocol, Virtual Society Client Protocol (VSCP), which runs over IP to send the changes in the users locations, and receive information about the location of other users.

CP uses a Client-sever communication model. All changes are sent to the server. The server uses the Area of Interest (AOI) algorithm to decide which users need to be aware of the changes. To overcome the problem of the server becoming a bottleneck, a replicated server approach has been developed. In this approach, each server is associated with an aura manager. The aura manager tracks objects and informs them of any aura collisions. A master aura manager provides consistency among servers. The communications among the servers is based on multicast.

The database is fully replicated in CP, and users must have a copy of the database before entering the virtual environment.

Community Place initially received much attention in the VRML community as the first browser to support multiuser features using its proprietary protocol. However, after the development of the External Application Interface (EAI) [SGI-EAI, 1996], which allows users to develop a multiuser environment using their own protocol, CP lost some of its popularity.

2.2.7 CALVIN

CALVIN (Collaborative Architecture Layout Via Immersive Navigation) is a system that allows participants to design collaboratively in a virtual environment [Leigh, July 1996]. It employs the heterogeneous perspectives, e.g. inside vs. outside, as a paradigm for collaboration.

CALVIN has been design to run in the CAVE environment [Krueger, 1991]. CAVE is a 10-foot cube constructed of three translucent walls that are rear-projected with stereoscopic images. Users wear LCD shutter glasses with magnetic trackers attached. They use 3-button wands to interact with the environment. The CAVE library can support several other display devices, as well.

In CALVIN the notion of providing two viewpoints for perspective viewing is called "mortals and deities" [Leigh, April 1996]. Mortals view the world from an "inside-out" perspective, while deities view world from an "outside-in" perspective. However, deities may assume more influential roles over mortals in a collaborative environment.

CALVIN maintains a central database, which contains a collection of objects, avatars and scene descriptions. Moreover, the database provides persistency, indirect collaboration and guarantees consistency across various environments.

CALVIN networking is based on the client-sever model. Client's are connected together via the centralised database. Other features of the system are speech recognition for menu selection and audio feedback. It also provides a Head-Up Display (HUD), called virtual visor, on which status information can be displayed.

CALVIN has been tailored for architectural design in CAD environments. It does not define an AOI for each user, and all modifications are resent to all participants. As a very strict consistency control is implemented in this system, the system cannot scale up. However, in these environments, the number of users is not high.

2.2.8 Spline

Spline (Scalable Platform for Large Interactive Networked Environment) provides an architecture for implementing multiuser interactive environments based on a shared world model [Barrus, 1996]. The world model is an object oriented database, which is distributed, and each user has a partial copy of it. To achieve low latency the consistency control is not strict in Spline.

The most prevalent type of object in the Spline world model is called a *thing*, and the central organising principle is termed *Locales*. Things have a hierarchical architecture, and each thing must be contained within one locale. The world model can be divided into many locales, whose addresses are the multicast addresses used for sending message about locales and things contained in them. Each locale can also have a parent.

As things move, their containing locales change according to their location in the virtual world. Moreover, update message for objects within a locale are sent to the appropriate address associated with that locale. This concept is very similar to the concept of AOI in the NPSNET system.

The other feature of Spline is that it does not have a global coordinate system. Instead, each locale has its own local coordinate system, and the position of each thing is specified solely with respect to its locale. This allows precise positioning and velocity, even with 32-bit floating-point numbers.

Spline does not provide any mechanism to control access to the objects. In addition, based on the published papers, it is not clear whether users can be actively involved and modify the scene objects, or whether they are passive and only watch the virtual world.

2.2.10 Discussion

The systems discussed in this section, although used for different applications, have one thing in common; they have tried to compromise between consistency, scalability and real-time operation. To highlight this compromise, important features of these systems have been compared in Table 2.1. This table shows that it is less likely to have a highly consistent, scalable and real-time DVR system. This fact has been stated by other researchers as well [Macedonia, 1996].

	Systems	Architecture	Database	Communication Method	Locking Mechanism	Consistency	Scalability
A	SIMNET	distributed	Fully replicated	Broadcast	NA	Loose	Medium
B	NPSNET	distributed	Fully replicated	Multicast	NA	Loose	High
C	DIVE	Semi-distributed	Partially replicated	Multicast	Explicit	Medium	Medium
D	AVIARY	Hierarchical	Central	NA	NA	Tight	Low
E	MASSIVE	Distributed	Fully replicated	Unicast	NA	Tight	Low
F	NVR	Client-server	Fully replicated	Unicast	Explicit	Tight	Low
G	CP	Client-server	Fully replicated	Unicast	NA	Tight	Medium
H	CALVIN	Client-server	Central	Unicast	NA	Tight	Low
I	Spline	Semi-distributed	Partially replicated	Multicast	NA	Medium	Medium
	WNMS	Client-server	Partially replicated	Unicast	Explicit	Tight	Medium

Table 2.1- Comparison between different DVR systems.

To visualise the difference between systems in terms of the database model, consistency and scalability, the 3D-surface graph of Figure 2.1 has been provided. The proposed system, Web-based Network Management System (WNMS), has been distinguished to show where that system stands in comparison with others.

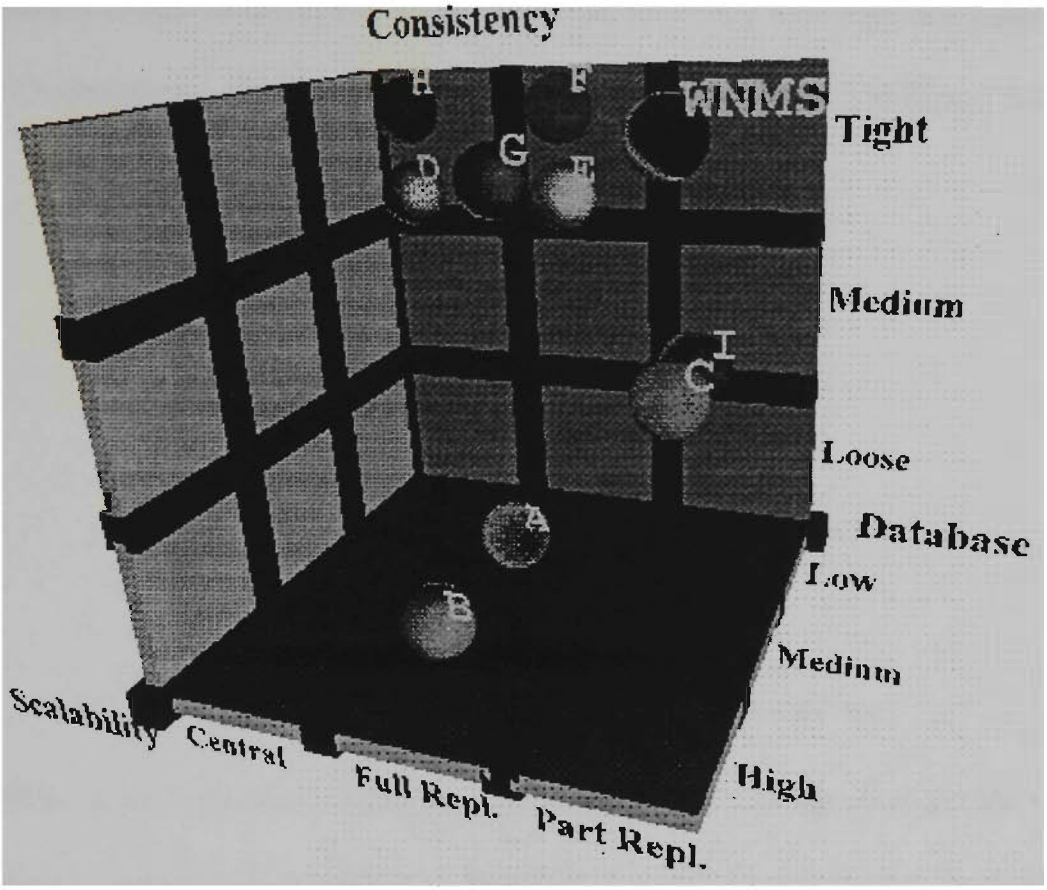

Figure 2.1- Comparing systems in a 3D-surface graph.

2.3 Non-traditional User Interfaces for Network Management

In this section, a brief review of several three-dimensional user interfaces is provided. In addition, some other non-traditional user interfaces for network management, such as bifocal display and Web-based user interfaces, are reviewed.

2.3.1 Columbia University's system

The Center for Telecommunication Research (CTR) at Columbia University was one of the first groups to exploit virtual reality for network management [Lazar, 1992]. They have designed and implemented a prototype system, which combines a Network

Management (NM) back-end and a VR front-end, to build a new interface paradigm for network management. The architecture of this system is reproduced in Figure 2.2.

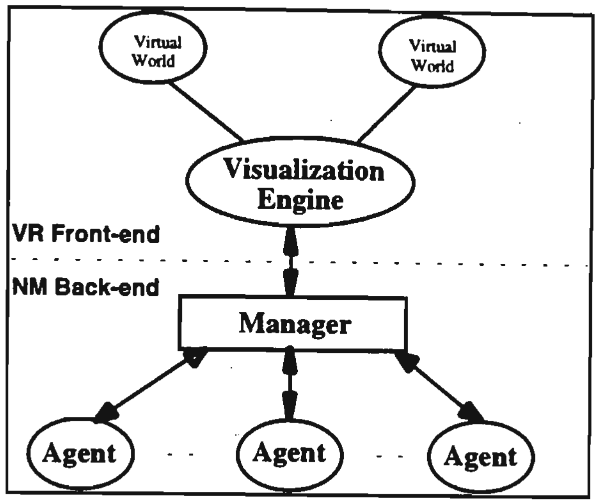

Figure 2.2- Architecture of University of Columbia system.

The NM back-end consists of a manager, which interacts with agents to get the status of the network elements. This system is based on the OSI model. It uses the OSI upper-layer provided by the ISO Development Environment (ISODE), and OSI Management Software provided by the OSI Management Information System (ISOMIS). ISODE supports the TCP/IP protocol, as well.

The information gathered by the NM part is sent to the VR front-end, which consists of a visualisation engine to build virtual worlds (VW) in which the user can monitor and control a live network. The VR part deploys a rendering system of 25000 polygons per frame, a VPL DataGlove and VPL EyePhone, which is attached to a HMD. Head position and movement is tracked by a magnetic sensor attached to the HMD.

The system does not support user manipulation of the system, and uses file-based communication between VR and NM parts. This system was later developed to work on

Broadband ATM networks. The modified system utilises a stereo LCD display, with interaction provided through a 3D mouse instead of the HMD and glove [Crutcher, 1993; Feiner, 1993]. This method combines the conventional 2D window system with a 3D virtual world.

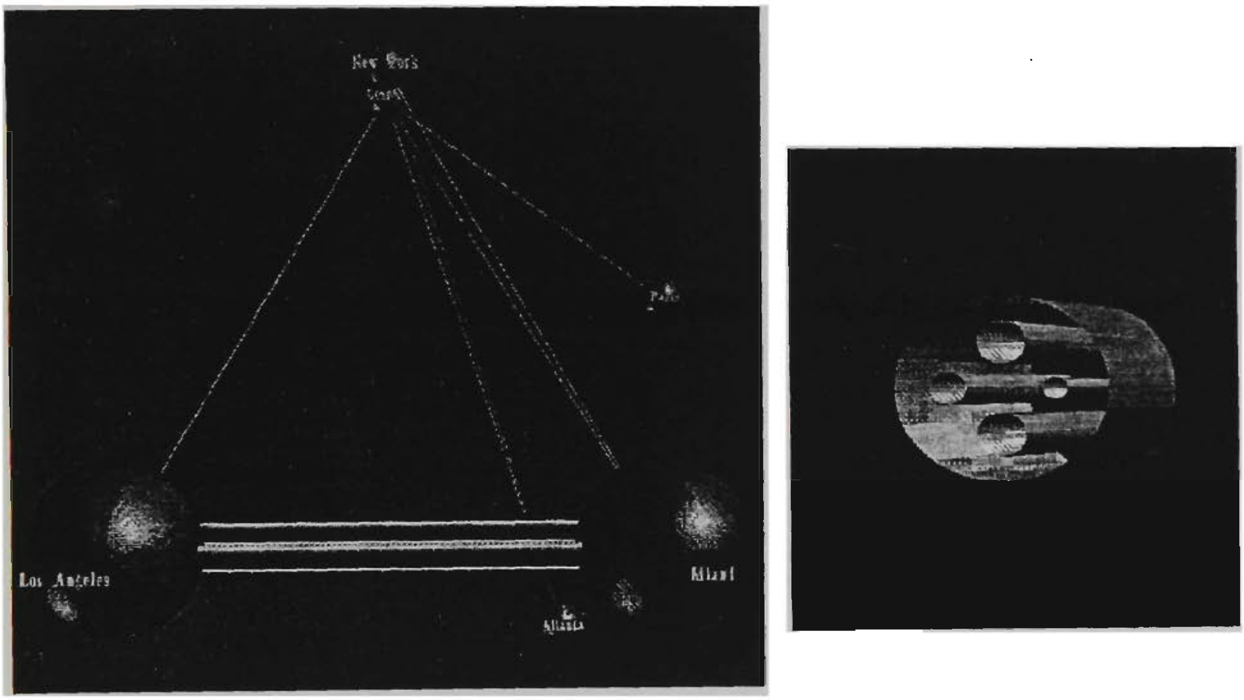

Figure 2.3- Inspection of a physical link

The latter system supports physical and logical topology views. In the physical topology view, nodes are presented as spheres and links as cylinders. The logical topology view consists of two views: a view of the virtual path (VP) on a single link or switch, and a view of the complete virtual path through the network. In these views, the diameter of each cylinder is proportional to the capacity of the corresponding VP. Statistics for the VP are displayed textually. The system displays and allows manipulation of basic physical and logical topology. Figure 2.3 shows how a physical link can be inspected in this system.

2.3.2 British Telecom's system

The researchers at British Telecom (BT) have also applied a 3D VR interface for network management [Walker 1993]. The system uses an interactive 3D desktop VR user interface to navigate through a network in real-time, moving between different perspectives and subsets of the underlying data. Some of the features incorporated into the system include layer selection and removal, context labelling, route selection, exception reporting and drill down [Rea, 1993].

Figure 2.4- Visual correlation of lightning strikes and network alarms [Walker 1995]

These features allow users to explore and interact with the network as a database and the results are displayed visually. Also, the system allows users with different capabilities (privileges), independently or by supervision of another user, to interact in a shared virtual world with different viewpoints [Rea, 1993].

The researchers have also applied VR technology to visualise the performance data in three-dimension. Figure 2.4 shows how lightning strikes and network alarms are correlated on a stormy day.

2.3.3 AT&T systems

AT&T has developed a suit of applications to visualise and analyse the strategic data sets and solve essential business problems [Eick, January 1996]. To overcome the problem of visualising large networking data, AT&T researchers have chosen three different strategies: dynamic parameter focusing, positioning and linked filters, and using a 3D layout [Eick, March 1996]. These three strategies are embodied in the three systems: SeeNet, NicheWorks, and SeeNet3D, respectively.

SeeNet allows a set of interactive adjustments so that the data become more meaningful to the user. The software supports seven classes of parameters that can be manipulated. The classes include statistic, colour encoding, thresholding, geography, time, size and line shortening. Several other parameters have also been suggested in [Becker, 1989] to generate a dynamic graph, or dynamic displays for networks.

NicheWorks has been developed to support those kinds of relationship that do not have a natural layout. The important factor, here, is the algorithm which put the related nodes near each other. An algorithm is presented in [Eick, 1993] for node positioning.

SeeNet3D uses the advantages of another spatial dimension. However, it restricts the display to a sphere capture as shown in Figure 2.5. Although it might avoid the problem of confusion, difficulty of navigation, and being lost in the 3D world [Eick, March 1996], it lacks the advantage of providing an intuitive immersive world.

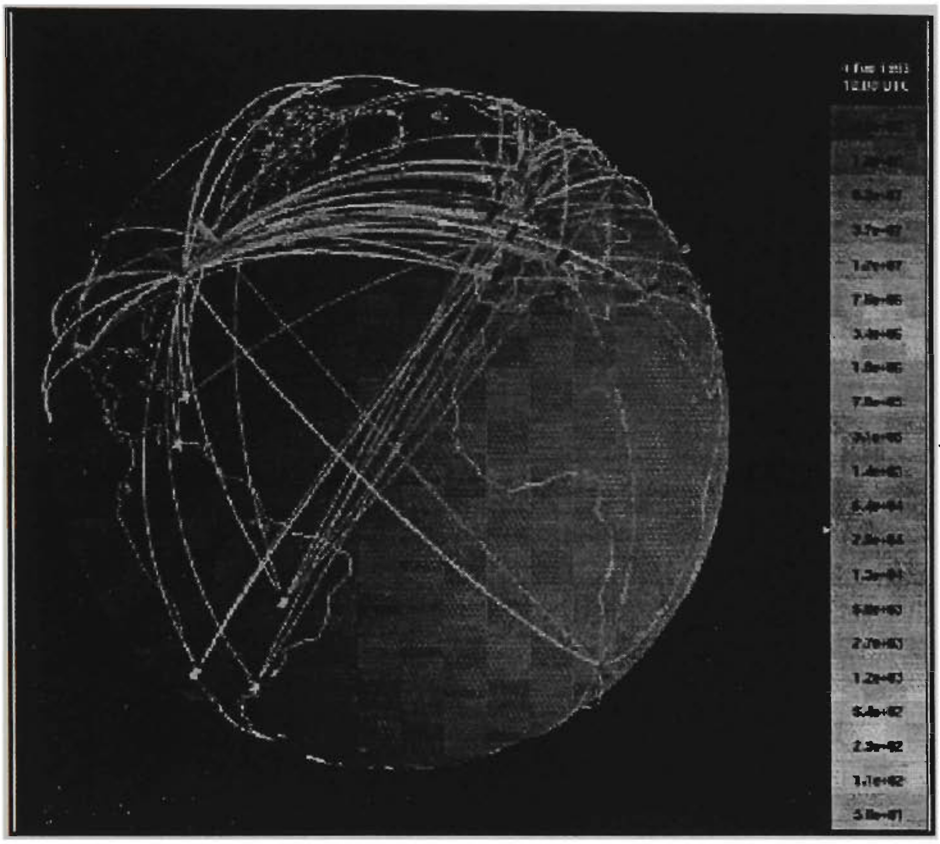

Figure 2.5-One frame from an animation showing worldwide Internet traffic over a two-hour period. [Eick, January 1996].

2.3.4 Unicenter TNG

Unicenter TNG has been built to provide complete end-to-end management of all computing resources that support critical business processes [Computer Associate, 1997]. It provides a 3D virtual reality display of physical and logical views of a network. Unicenter TNG has facilities for end users to easily navigate throughout the global IT environment, for IT managers to manage the entire IT environment, and for IT administrators to address business-relevant questions. A web-based user interface has been incorporated into the system, though it is not yet publicly available. Figure 2.6

shows a real world interface of Unicenter TNG system. Unicenter TNG can be considered as the first commercial VR based user interface for network management. Thus demonstrating that the need for better visualisation methods is being acknowledged by the company as well.

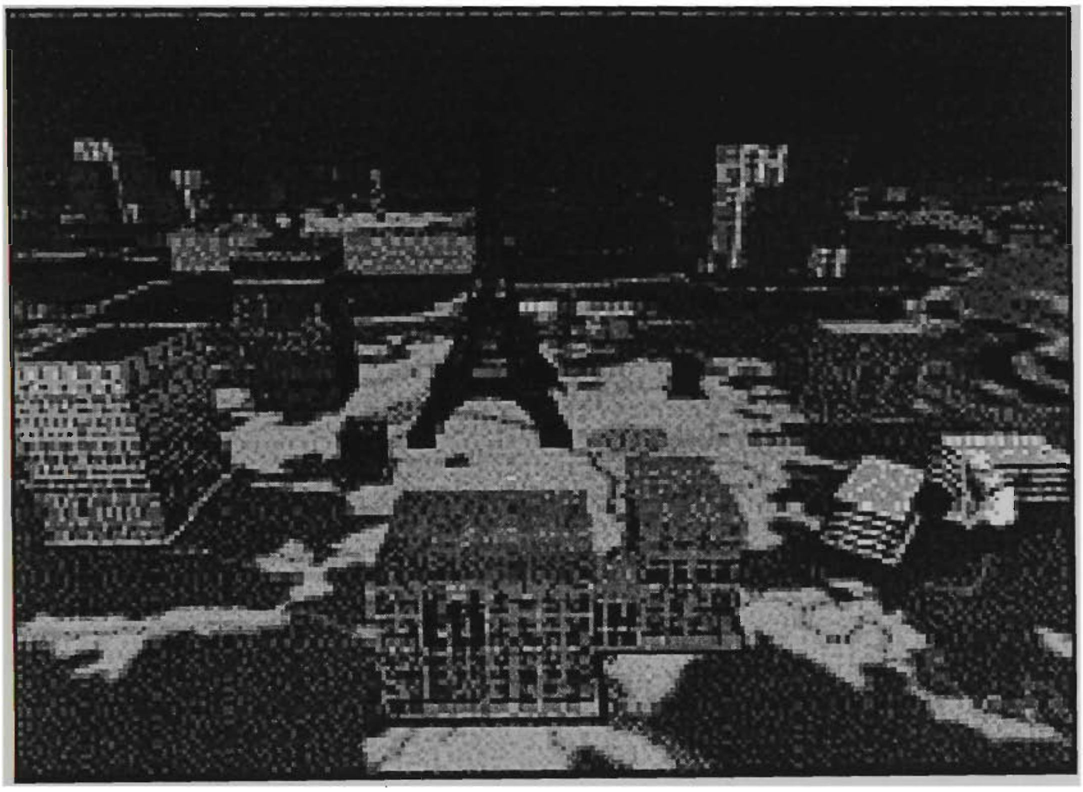

Figure 2.6- Unicenter TNG real world interface.

2.3.5 Bifocal user interface

Bifocal or fisheye display is a method if displaying a large amount of data within a limited area [Sarkar, 1992]. In this method, a single window has been divided into nine regions; one of them will be enlarged based on the interest of the user in that region (Figure 2.7). The advantage of this method is that all objects can be displayed without window overlapping. So, the user can focus on a region of the display, while maintaining an overall view of all objects in the window.

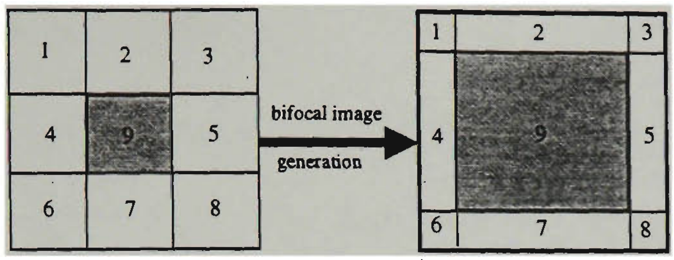

Figure 2.7- Bifocal displays window regions

The characteristics of bifocal display makes it a good candidate for network management. However, because of the high processing requirement, a hardware system is required for real-time visualisation. Fuji has described how a bifocal display boosted by hardware system can be used for network management [Fuji, 1994]. The system has been used for provision of event monitoring. Whenever an alarm occurs, the related symbol starts blinking. By clicking on the region of the alarm, the region is enlarged, and other regions are compressed. After recognising the location and the reason for the alarm, the operator initiates the appropriate processes to fix it.

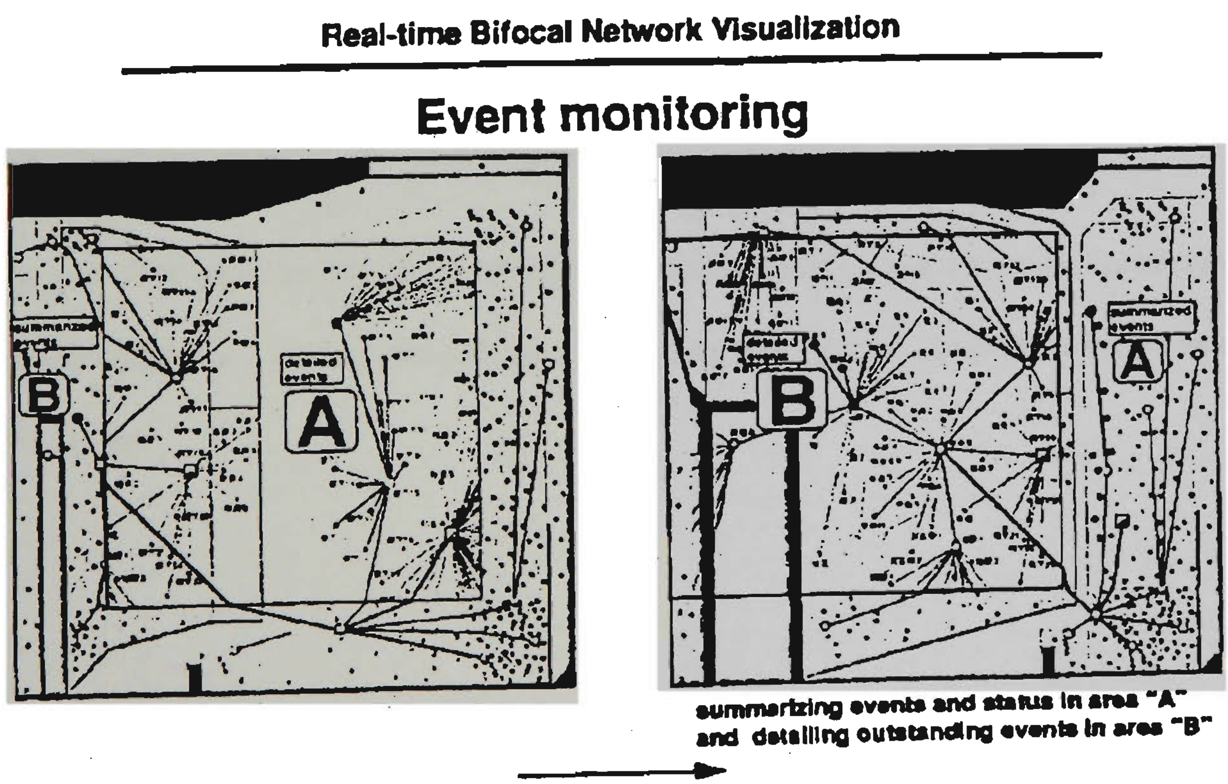

Figure 2.8- Moving the details in a bifocal display [Fuji, 1994]

2.3.6 Web-based user interfaces

The rapid growth of the World Wide Web has made it a good candidate for several applications. The initial deployment of the Web in network management was in network element management. That is, using HTTP instead of SNMP, or in addition to it, for device polling. In this regard, several Internet draft documents were developed. Web-Based Enterprise Management (WBEM) [WBEM Consortium, 1996] and HTTP-Based SNMP and CMIP Network Management [Deri, 1996] and Web Based management [Harrison, 1996] being among them. In addition, Java Management API (JMAPI) provides a new way of communication with network elements [Sun JMAPI, 1996].

Many companies have incorporated Web capability into their devices and network management systems [Jander, 1996]. In addition, some companies have started to use Web-based user interfaces for network management. For instance, in the second half of 1996, Bay Networks' Optivity Web product and Computer Associate's Unicenter TNG announced incorporation of a Web-based user interface as part of their systems.

Optivity Web provides 2D graphical HTML-based monitoring and viewing capabilities to Bay Networks' Optivity network management system. Figure 2.9 shows a view from that system.

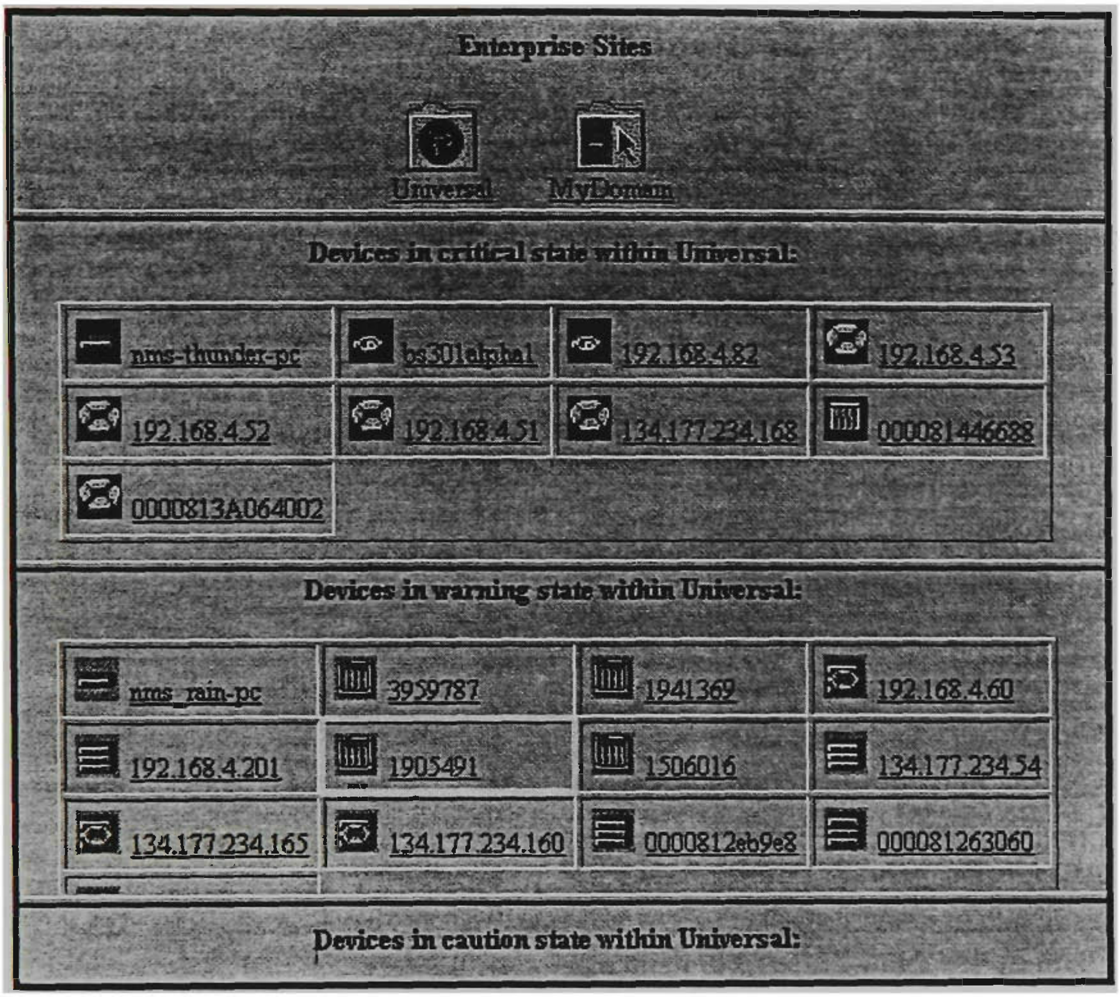

Figure 2.9- A typical view of Optivity Web product.

In addition to these systems, researchers in IBM Zurich research laboratory have applied VRML to visualise network management resources [Deri, 1997]. In this system, VRML has been used to provide three functionalities: 3D topological view of the network, hierarchical information, and compound information view.

This system has been applied to ATM technology. For viewing the logical topology of an ATM network, nodes are shown as spheres and links as cylinders (see Figure 2.10). To indicate which sites are the main ones, their corresponding symbols have been placed vertically higher than others. In addition, each node has an HTML anchor, which if clicked, shows detailed information about it.

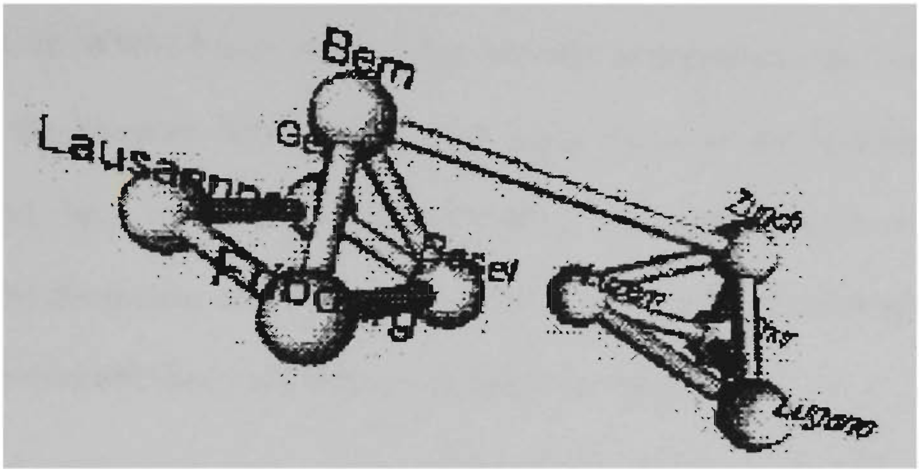

Figure 2.10- VRML-based ATM logical topology [Deri, 1997].

To show the hierarchical information and compound views, the system uses the LOD (Level Of Details) feature of VRML. This means that elements far from the viewer are shown as primitives. Moving nearer to the objects, a detailed representation of them can be viewed. Using this method, all objects are placed within one VRML file, without slowing the rendering rate; objects are only shown if the viewer is near enough to them.

The problem with this method is that it does not scale up. As the number of network elements increases, the size of the VRML file becomes too big to fit in the memory. In addition, it takes significant time to download the file, as VRML does not have any load-on-demand mechanism.

2.3.7 Others

The GEC-Marconi Hirst Research Center is also using VR as an interface for telecommunication systems [Stanger, 1992]. Their prototype system is interfaced to VPL EyePhones and Division's 3D mouse and hosted by a PC. A SUN SPARCstation has been utilised to provide texture mapping. The latter system uses a 2D mouse for viewpoint navigation.

Another set of WWW-based systems for network management has been reported recently in the literature [Chen, 1996]. The paper discusses the implementation of HTML based, Java based and VR based systems for ATM network management. Following the discussion, the paper concludes that the combination of VR and WWW systems improves efficiency and reduces the cost of investment.

Other research regarding 3D user interface for network management has been undertaken mostly by private telecommunications companies. For example, Optus Australia and NTT Japan are known to be working on such user interfaces. However, little information is available in the literature about the methodologies and the kinds of user interfaces used by these companies.

2.3.8 Discussion

Three-dimensional and web-based user interfaces for network management have attracted much interest in both academic and commercial organisations. The three-dimensional user interface for network management was initially introduced by Columbia University in 1992. However, as the price of VR equipment decreased dramatically, it became affordable for more academic and non-academic research groups to work in this area. The introduction of VRML in August 1994, and its rapid growth to become a standard (Version 1.0) in May 1995 [Bell, 1995], paved the way for a generation of portable networked VR systems.

The evolution of the Internet and ubiquitous use of the World Wide Web (WWW) encourages researchers to think of using this technology to provide a universal user interface for network management. Therefore, this field attracted many researchers, independently working in the area. Although the focus of each researcher might be different from the others, similar concepts can be found among their prototype systems.

2.4 User Interface Evaluation

The aim of this section is to provide a review of the available evaluation and measurement methods. In addition, a comparison is made to reveal which methods are suitable in a particular context. Some of these methods are used in Chapter 6 to evaluate WNMS and compare it with traditional user interfaces.

2.4.1 Evaluation methods

Evaluation is the task of gathering information about the usability of a system for a specific group of users in a particular environment context and for a specific activity [Preece, 1995]. There are several reasons for doing evaluation, such as comparing designs, understanding the real world, checking conformance to existing standards, and engineering toward a specific target. The selection of methods of evaluation depends upon the aim of the evaluation and its context.

2.4.1.1 Observation and monitoring

In observation and monitoring methods, the interaction of the users with the system is monitored, either directly or indirectly. In direct observation, the evaluator observes the subjects as they work with the system. As it is possible for the evaluator to overlook some important things, or the user's inability to act naturally due to the presence of the evaluator (Hawthorne effect), the observation is sometimes done using audio and/or video recording. Indirect observation, although repeatable, is usually difficult to analyse, because of the high volume of data [Preece, 1995].

Data may also be collected by keystroke logging. In this method, either the system is changed or some utility software is used to log the user's keyboard input. It is possible to attach a timestamp to all keystroke or log the interaction of the user with different sections of the system.

2.4.1.2 Collecting user opinions

Although objective measurement provides useful information about the performance of a system, it is desirable to find out what users think about the system, too. Collecting user opinions can be done with interviews or questionnaire. Interviewing can be structured with predetermined questions or flexible with just a set of topics with no predetermined sequence. Flexible interviewing can result in a better knowledge elicitation, but is difficult to achieve and depends on the experience of the interviewer [Welbank, 1990].

Questionnaires and surveys, on the other hand, focus on preparing unambiguous questions rather than flexibly gathering data. There are two kinds of questions: closed questions and open questions. In closed questioning the respondent should select one (or some) of the alternatives explicitly mentioned in the question. For this method, there are several rating scales, each of them suitable for a particular situation [Briggs, 1987]. In open questioning, the respondent is free to express his views. Regardless of the method, collecting user opinions is an important part of system evaluation.

2.4.1.3 Experiment and benchmarking

In this method, well-designed laboratory experiments are used, and all variables of interest are controlled. In traditional experiments, a hypothesis is stated and all independent and dependent variables chosen to test the hypothesis. Then, subjects are selected so that they are a good sample of the whole population of the users of the system, and the experiment is performed. Statistical techniques are used to analyse the data and generalise the result.

As controlling the variables is usually difficult, an engineering approach known as usability engineering has been adopted [Whiteside, 1988]. Usability engineering uses metrics and the usability specification that are defined for each task, and deploys benchmark tests that are given to users in a semi-scientific condition. The user's opinion is also gathered through interviews and questionnaires to produce attitude metrics. Everything related to usability specification is recorded and is used for quantitative analysis producing metrics to guide system design.

The problem of usability engineering is that it is usually too expensive to provide the required environment. Therefore, a discounted method is usually used [Nielsen, 1989]. Discount usability engineering is explained later as an inspection method.

2.4.1.4 Interpretive evaluation

The purpose of this method is to better understand the behaviour of the users under natural circumstances. The data is collected informally, so the result is a qualitative evaluation of the system, and the analysis is more difficult than formal methods. In order to elicit more information, the cooperation, and even participation, of the user in both data gathering and analysis phases is required [Monk, 1993].

2.4.1.5 Predictive evaluation

Unlike other methods, predictive evaluation does not need to involve the actual users of a system. In fact, these kinds of evaluation can be done even before the system is implemented. Predictive evaluation consists of two sets of methods: inspection methods and modeling methods. Inspection methods involve the evaluation of the system by specialist and experts who are aware of the technology of the system and have some experience in HCI [Preece, 1995]. Modeling methods employ psychological modeling techniques such as keystroke analysis to predict the usability of the system. In keystroke analysis, the time to execute a task is broken into several components, such as keystroke time, response time, etc, and by analysing them, the usability of the system is predicted [Card, 1983].

Inspection methods consist of several techniques. Usage simulation involves reviewing the system by experts who simulate the behaviour of novice users to find out the usability problem that the less experienced user may encounter [Hammond, 1984]. Structured expert reviewing techniques are the same as usage simulation, but the reviews are carefully planned and structured.

There are three different structured expert reviewing methods: heuristic evaluation, discount usability engineering and cognitive walkthrough. In heuristic evaluation the expert inspection is guided by a set of high-level heuristics, which help them focus on key usability issues of concern [Nielson, 1992]. Each reviewer goes through the interface several times to inspect the flow of the interface from one screen to another, and to inspect the features of each screen against heuristics.

Discount usability technique enables developers to benefit from the facilities of usability testing even if they have few resources [Nielson, 1992]. It is a hybrid of usability engineering and heuristic evaluation, which relaxes some of the restriction of usability engineering, in order to save time or money.

The third structured expert review method is cognitive walkthrough [Wharton, 1994]. In this method, firstly, experts define the exact task, its context and the assumption about the expertise of the users. Then they walk through the task and review the action that the user should do to achieve the task. The experts try to predict user behaviour and the problems that they might encounter. The advantage of this process is that it detects problems in early stages of design.

2.4.2 Measuring techniques

In order to provide quantitative values for various aspects of system usability, some important parameters of the system have to be measured. Here, several measurement techniques, developed for measuring different aspects of systems, are discussed.

2.4.2.1 Performance measuring

Performance analysis measures the efficiency and effectiveness of a system. The behaviour of the user is monitored either directly or indirectly to detect the usability problems of the system. Some important factors are proportion of task completed in a specific time, and the quality of output, that is number of error and mistakes. As mentioned above, indirect observation has several advantages over direct monitoring, but the analysis of video and audio tapes is difficult and time consuming. To overcome the problem, a software tool, called DRUM (Diagnostic Recorder for Usability Measurement) [Macleod, 1993], has been developed for the MUSiC (Metrics for Usability Standards in Computing) project of ESPRIT programme.

DRUM helps analyst build a time-sequenced log of events, which speeds up the analysis of video tapes. Once an observed event is logged, DRUM can locate that event and replay it, and from the log of sessions, it can calculate usability measures for each subject. Data prepared by the software also can be exported to statistical packages for more analysis. However, due to the high amount of resource requirements, DRUM is only suitable for big projects.

2.4.2.2 Cognitive workload measuring

It is usually desirable to estimate the amount of effort the user invests to perform a task using a system. Obviously, if good performance can be achieved only at the cost of high invested effort, the system has low usability [Corbett, 1993]. Cognitive workload measurement can be done either objectively or subjectively. In objective measurement, the users' mental effort while performing the task is measured using their heart rate variability and their respiration rhythm [Wiethoff, 1991].

Subjective measurement of the cognitive workload can be done using questionnaires that ask the user to rate the mental workload they have experienced in using the system. Among several developed questionnaires Subjective Mental Effort Questionnaire (SMEQ), by WITlabs at University of Delft, and TaskLoad Index (TLX)[Biferno, 1985], by NASA Ames Research Center have been widely used and internationally acknowledged. It also has been shown that NASA-TLX is superior to other methods in terms of sensitivity and operator acceptance [Hill, 1992].

2.4.2.3 User attitude measuring

The feeling of the users in respect to a system is measured using interviews and questionnaires. SUMI (Software Usability Measurement Inventory) is an international standard questionnaire used to determine the usability of a software [Porteous, 1993]. Users complete questionnaires after using a system. SUMI gives a global usability score, which can be used for comparison of the usability of user interfaces. Moreover, it provides some measure on the following criteria: efficiency, affect, helpfulness, control and learnability.

Overall, SUMI has 50 questions, each of which can be answered as 'agree', 'undecided' or 'disagree'. In order to have a significant result, at least 10 users should complete the questionnaire.

The problem is that SUMI is helpful if there are enough statistics about similar systems. That is, the global score that SUMI produces should be compared with the score of a similar system to rate the system. For instance, if there are not enough data on SUMI database about spreadsheet software, it is not possible to use SUMI to evaluate a specific spreadsheet program.

2.4.2.4 Analytical measures

Analytical measurements are usually performed early in the development of user interfaces. They are based on the dynamic modeling of user interfaces, and on models of user tasks. The Skill Acquisition Network (SANe) toolkit [Gunsthoevel, 1991] has been developed within the MUSiC project to analytically predict the usability of the system. Using this toolkit, 25 measures that describe the different aspects of usability quality are performed. These measures include: efficiency of use, learning requirements, cognitive workload, adaptedness of task and robustness.

In addition, a quantitative description language for user interfaces, as described in [Rauterberg, 1993], has been developed to formally measure some aspects of user interfaces, such as 'amount of feedback' or 'interactive directness'. In this model, interactive space is divided into function space and object space; each of which has a set of hidden and perceptible representations. Estimation of different aspects is undertaken by calculating the average number of functions and objects required to achieve the tasks to fulfil that aspect.

Another analytic measurement method is GOMS (Goals, Operations, Methods and Selection Rules) model [Card, 1983]. GOMS is a formal analysis of the interaction of a user with the computer that estimates the performance of an expert user regardless of learnability problems.

GOMS models may be constructed after the design or implementation of a system or during the design. In addition, it should be mentioned that development of the GOMS model for more interactive systems would be more difficult.

2.4.3 Comparisons of methods

In previous sections, different methods and techniques were described for user interface evaluation. It is important to note that each of these techniques is suitable for different phases of the life cycle of product development. For instance, predictive methods can be applied in early stages of product development, while for experiment and benchmarking methods, a working prototype of the system should be available. Additionally, each method only evaluates particular aspects of a system. Table 2.2 summarises the relationship between different kinds of evaluations and the purpose of evaluation [Preece, 1995].

	Observing & Monitoring	Users' Opinion	Experiment & Benchmarks	Interpretive	Predictive
Engineering toward a target	✓	✓✓	✓✓	✗	✓
Understanding the real world	✓✓	✓✓	✗	✓✓	✗
Comparing designs	✓✓	✓✓	✓	✗	✓✓
Standard conformance	✗	✗	✓✓	✗	✗

Legend:
✓✓: More Likely
✓: Less Likely
✗: Inappropriate

Table 2.2- Relationship between different kinds of evaluation and the purpose of evaluation.

Several researchers have applied different techniques for usability evaluation in various environments. In [Jeffries, 1991; Miller, 1992] four techniques: heuristic evaluation, guidelines, cognitive walkthrough and usability testing are compared with each other in terms of the number of usability problems, their severity, and the methods found in a particular product. The authors report that heuristic evaluation found most problems and identified more serious ones.

In another experiment, three measuring techniques: cognitive workload, performance analysis, and psychometric measures (SUMI questionnaire), were used to compare MS-Word 5.0 (DOS) package with MS-Word 1.1 (Windows) [Houwing, 1993]. These techniques have been used supplementarily to measure different aspects of the usability of the systems.

2.4.4 Discussion

In this section, evaluation methods were briefly discussed. As shown in Table 2.2, each discussed method is suitable for a specific purpose. That is, each method examines a different aspect of a system. A comprehensive evaluation usually requires utilising several methods. For instance, one might use predictive methods, including heuristic evaluation and comparison against guidelines, to provide a quantitative indication of system performance. User opinions can then be collected to reveal the amount of satisfaction of the users with the system.

2.5 Summary

In this chapter, a review of the literature regarding Distributed VR system, non-traditional network management system and rigorous evaluation and comparison methods was discussed. Firstly, several distributed virtual reality systems were described. The systems included works in NPSNET, DIVE, MASSIVE, AVIARY, CALVIN, Community Place, NVR system and Spline. Each of these systems has made several assumptions and trade-offs, which makes them appropriate for the environment for which they were intended to. However, none of them addressed the issues related to multiuser interfaces.

Several non-traditional user interfaces for telecommunication network management were also discussed. Among 3D user interface systems, Columbia University, British Telecom, AT&T Bell Laboratories and Unicenter TNG systems were discussed in some detail. In addition, bifocal display user interfaces and web-based user interfaces were discussed as some non-VR non-traditional user interfaces.

A review of evaluation and measuring methods was also presented. Beside describing different methods, context in which each method can be applied was shown. This information is used in Chapter 6 to evaluate the Web-based network Management System (WNMS) and compare it with a traditional user interface for network management.

Chapter 3

Immersive Virtual Reality User Interface

3.1 Introduction

Networks are becoming increasingly more complex and automated [Rea, 1993]. This is due in part to their increased bandwidth, the need for automated protection switching, the deployment of virtual networks based on B-ISDN and ATM technology and the increase in traffic complexity typified by multimedia communications. In turn, network management is becoming more complex and more mission critical to a larger number of organisations. In the past, this has led to the development of integrated network management systems using Windows Icons Mouse Pointer (WIMP) based direct manipulation user interfaces.

To reduce the risk of human mistakes and help the manger find the optimum solution, expert systems for alarm and fault correlation and diagnosis were then integrated with the management systems. These integrated network management systems have attempted to increase the ability of network operators to keep pace with the increased size and management demands of modern networks.

To further enhance the network management operating environment the use of Virtual Reality (VR) user interface technology for network management applications has been proposed [Crutcher, 1993; Kahani, 1995]. This work is important because it will lead to more natural network management system interfaces and, hopefully, to further increases in network operator productivity.

In Chapter 1, some of the limitations of traditional user interfaces were discussed. VR technology can address some of these limitations. VR interfaces consist of realistic 3D real-time animated displays, 3D interaction devices and 3D audio as shown in Figure 3.1. The main advantage of 3D visual display is its additional spatial dimension. In addition, using a Head Mounted Display (HMD) provides more immersion, which increases the semantic level of the display.

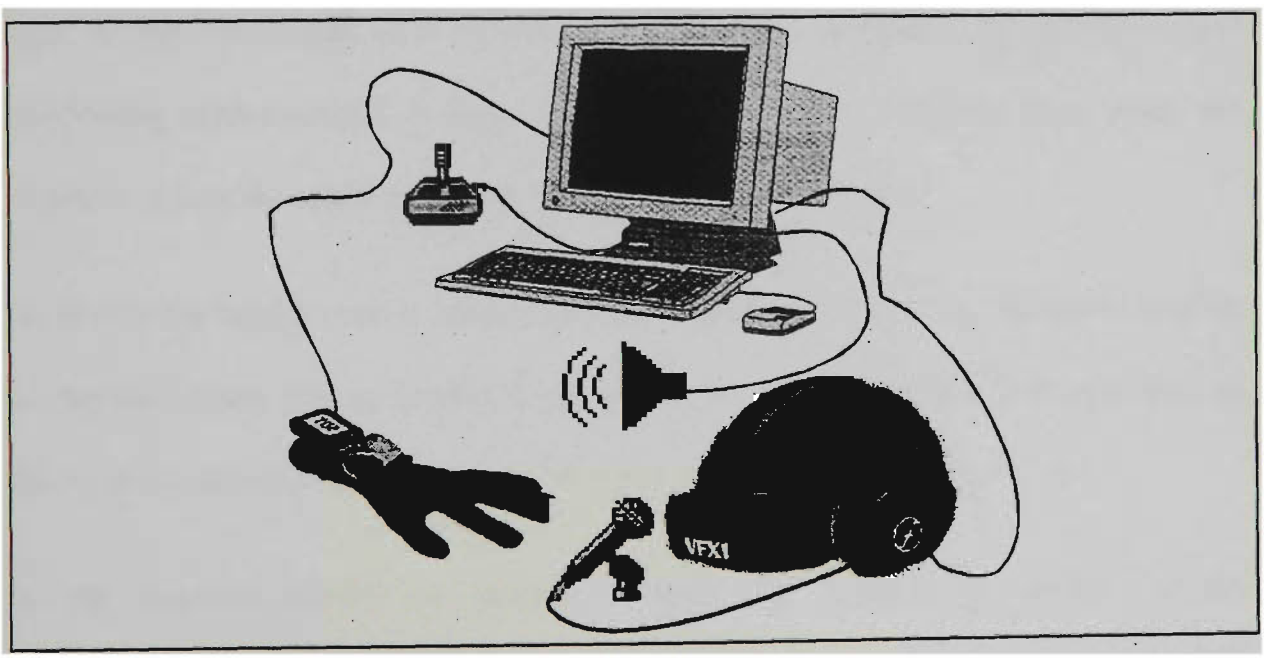

Figure 3.1- Some of the elements of VR systems.

Three-dimensional audio also adds another dimension to the display system. Research has shown that VR interfaces that do not seem realistic, because of simple graphical icons or low frame update rate, are more immersive if incorporated with 3D sound [Krueger, 1991]. Using a text to speech converter and 3D sound localisation, objects can represent some of their data using sound rather than text. All these together, generate an immersive environment in which the operator can perceive network elements and their behaviour more naturally.

The other element of VR interfaces is three-dimensional input devices. They improve the level of interaction, which is another essential factor for user immersion. Using a VR glove, the user interacts with the virtual world directly by manipulating the objects in the same way they would be manipulated in the real world; potentially providing higher levels of immersion and interaction than WIMP interfaces.

With a speech recognition system, sound can also act as an input device. That means, most simple commands, such as changing view point, navigating up and down, and confirming system actions, is done by talking. This, firstly, speeds up these tasks, and secondly, keeps the operator's hands free for more important tasks.

As HMDs are usually used in conjunction with a head tracker system, the head's rotation and/or movement, acts as another input device. The change of viewpoint according to the head movement simplifies the task of navigation to a basic natural action.

A VR interface allows the network structure, its performance levels and the management information flows to be visualised together, potentially in real-time. This provides more extensive interaction than currently envisaged by the TMN g interface [Sahin, 1994].

In this chapter, an immersive VR user interface for network management is discussed. Firstly, the architecture of the system is explained. This is followed by the design methodology of the user interface. Then, the method used for network visualisation is discussed. Implementation issues, observation and a qualitative evaluation of the system are also presented in this chapter. Finally, in the conclusion, the reasons for moving toward a desktop VR, rather than immersive one, are discussed.

3.2 System Architecture

The system has two main parts: the user interface and the network management. In order to focus on user interface issues, and not to engage in low-level device polling activities, an existing network management system was utilised for the network management part. As shown in Figure 3.2, these two parts communicate with each other via the Command Line Interface (CLI) of the Network Management System (NMS).

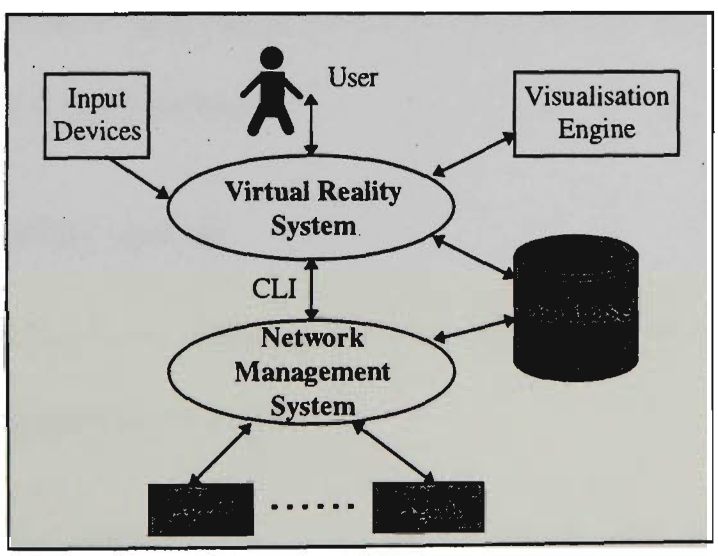

Figure 3.2- The architecture of the system

3.2.1 Network management system

Cabletron SPECTRUM was chosen to act as the core NMS. There are several reasons for this selection. First of all, Spectrum itself has a similar structure to that of the proposed system; it consists of a server (SpectroSERVER) and a graphical user interface (SpectroGRAPH). This simplifies not only the implementation, but also the comparison between the VR-based user interface and Spectrum's two-dimensional WIMP-based user interface, SpectroGRAPH.

The other reason is that Spectrum is one of the better network management systems. Therefore, one could rely on it for reliable device polling, alarm detection and event correlation. In addition, as the system was already installed, and was in use, local expertise was available for qualitative and quantitative interface assessment.

SpectroSERVER uses the SNMP protocol for device polling and alarm detection. Although, it is possible to define other protocols using the Spectrum toolkit [Spectrum Manual, 1995], it is not an easy task.

3.2.2 Virtual reality system

The VR system has the basic virtual reality elements such as 3D image rendering and 3D navigation tools as shown in Figure 3.1.

Input devices used in different stages of the system development consist of a Logitech three-button mouse, Logitech Cyberman, a two-button joystick, and a 5th Dimension Glove. The output devices consist of a 15" Monitor, and a Forte VFX1 head mounted display. In the early stages of development, a pair of VR glasses that provided three-dimensional viewing the screen was also utilised. Detailed information about VR devices used in this system can be found in Appendix B.

3.2.3 Communication between VR system and NMS

Three pieces of information are retrieved from the network management system:

1. Network configuration.
2. Network topology.
3. Performance/fault data.

Network configuration information is changed when a device is added, deleted, or some of its characteristics changes, e.g. software upgrade. Network topology information contains the connectivity information. Both are nearly static and rarely need updating. Therefore, they are extracted, as network topology changes, from the NMS using its Command Line Interface (CLI), and automatically, a virtual network world database is constructed. This database is used by the VR system to build the virtual environment.

On the other hand, the performance and fault data are quite dynamic, requiring continuous updates. To provide a real-time user interface, performance/fault data must be collected directly from the NMS. This could be achieved by establishing a direct link between NMS and VR systems, in which the VR system sends its inquiries to the NMS using CLI commands to retrieve the required data.

3.3 User Interface Design

As the user interface is the focus of the thesis, its design requires special attention. The user interface should be flexible, configurable, scalable, and possibly portable to other platforms. It should be possible to configure the user interface to work with different input and output devices. Moreover, it is desirable to be able to add new I/O devices. Portability was also an important issue, so that the system could be used in other platforms with minor modifications.

To address these issues, an object-oriented approach was considered, and C++ selected as the programming language. The advantages of C++ are that it could provide a mixed environment of conventional and object-oriented programming, which was required for linking with graphical libraries. In addition, C++ compilers are available for most platforms. In this project although the Borland C++ compiler was used, the program was written so that it could be easily ported. Consequently, later, most of the code was compiled under Sun Solaris 2.3 with little modification.

Because of the object-oriented design, each element of the interface was designed to be an object, which can communicate with others by sending or receiving messages. The following major object classes were considered:

- *InputDevice* class, which currently includes keyboard, joystick, mouse and glove.
- *OutputDevice* class, which includes monitor and HMD.

- *World* class, which is the environment in which objects are present.
- *NetworkElement* class. Different types of network elements are subclasses of this class.
- *Connection* class, which connects network elements together.

The system has a main loop that constantly updates the virtual world based on the user input. This loop is shown in Figure 3.3 in pseudo-code format.

```
Loop:
    Check user input
    If nothing has changed
        Continue
    Draw the updated world in the shadow screen
    Show the shadow screen   // to make the screen flicker-free
End Loop
```

Figure 3.3- The pseudo code of the main loop

A more detailed explanation of the classes is presented in Appendix A. The relationship between these classes using Object Modelling Technique (OMT) graphical notations [Rumbaugh, 1991], is shown in Figure 3.4. Note that in this figure, only the important attributes and operations of the classes are shown.

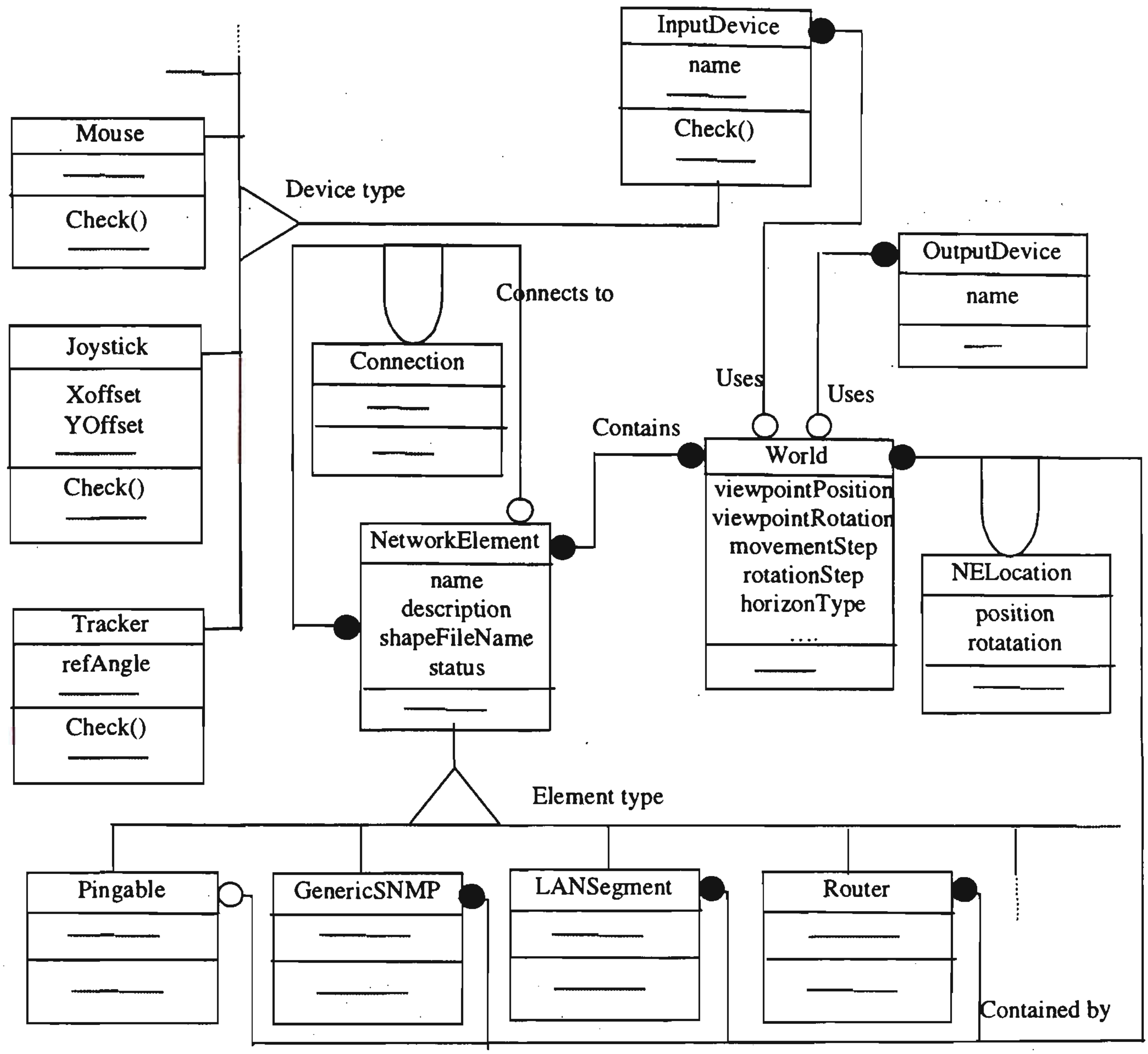

Figure 3.4- OMT notation for defined classes and their relationship

3.4 Visualisation Methods

There have been much research on 2D and 3D visualisation techniques [Batini, 1993; Ribarsky, 1994]. However, most of it concentrated on visualisation of data residing in a database. This data usually does not have any intrinsic physical or logical representations in the real world. Some are even abstract concepts. This fact led the researchers to try to fit different visualisation techniques to the data, to see which method fitted better.

For instance, the *building* metaphor is one of the popular visualisation metaphors, because of its hierarchical architecture. Catarci et al. used this metaphor to visualise the contents of a song database [Catarci, 1995]. A multi-storey building represents the whole database, and each floor is allocated to a specific music. Within each floor, there is one room for each singer, which is accessible through the corridor. Entering the room, users see different objects and furniture. The furniture contains different drawers, each storing one CD and its related information. Finally, at the lowest level, data related to each song is stored in folders within the drawer.

However, visualising computer networks are different to these methods, as they already have an intrinsic physical and logical structure in the real world. Either the physical or logical structure or a combination can be used for visualisation. Systems mentioned in Chapter 2 have used similar approaches. In fact, in order to provide a realistic immersive system, the visualisation should resemble the representation of the network in the real world, so that the users can quickly recognise the network elements. The logical structure, that is, how computers are connected together not where they are physically located, was chosen for the prototype system, because it provides more information about the network.

Initially, it was decided to show all network elements within the domain, e.g. the whole campus network, in one view. However, when implemented, several problems arose.

First, because of the large number of network elements, it was confusing and cumbersome for the user to find a particular network element. In addition, the user was often lost in the virtual world, and had to reset back to the original view. The high processing time for rendering a big network was also another factor. A large number of objects within a view causes the frame rate (number of rendered frames per second) to drop dramatically. In practice, it was found that frame rates less than ten have unsatisfactory effects on the feeling of immersion. In addition, the method, obviously, was not extensible. As discussed in Chapter 2, different LODs (level of Details) could speed up the rendering, though could not provide extensibility.

Consequently, the virtual world was broken into smaller worlds connected together via portals. The breakdown was based on the logical structure of the network. For instance, each LAN segment is shown in one world, and is connected to other parts of the network via a bridge or a router (similar to their real-world counterparts). In this case, for example, the bridge or router acts as the portal. In order to go to another world, the user should navigate into a portal.

This method proved to be more practical. The virtual worlds become much smaller, and there was less chance of the user being lost or confused. Also, as the rendering of each virtual world required less processing time, they were rendered at a higher rate. This approach (dividing into smaller worlds) has been used by several systems, such as DIVE [Carlsson, 1993] and MASSIVE [Benford, 1994].

The drawback of this method is that the whole hierarchy of the network cannot be observed explicitly in one view. This problem can be solved by providing a less detailed view of the whole network at the highest level. For actual monitoring and management of the network the user has to enter smaller, but more detailed worlds. This method was implemented for the second system discussed in Chapter 5.

3.5 Implementation Issues

The PC platform was chosen for the prototype implementation. There are several reasons behind this selection. First of all, most input/output devices, such as HMD or gloves are available for the PC. Using them on other types of computers is either not possible or requires writing a new device driver, which is difficult and time-consuming. Also, for the PC platform there are some inexpensive graphical libraries that can used to build a prototype system. These libraries for other platforms were scarce and expensive when the project started, but currently more tools are available for UNIX and Apple computers.

It was decided to start with an off-line system for the first prototype to examine the suitability of the system, and move to an on-line system and add some other features, e.g. multiuser, if the results were satisfactory. So, the network configuration and its status were extracted from the network management system and stored in several files. The system reads those files and the configuration file at the startup, to setup the virtual world and to determine which input/output devices are available.

The shapes of 3D symbols representing network elements are also important. These symbols should be defined so that by looking at them, the user could quickly recognise them. It should be noted that using the shapes of the objects in the real world as the objects' symbol might not always be useful. For example, bridges, routers, and simple pingable devices might look similar in the real world. The difference between these devices stems from their different functionality, not their different appearances.

In addition, it should be considered that complex symbols consist of large numbers of polygons. As the frame rate decreases when the number of polygons in a virtual world increases, complex symbols will adversely affect the rendering rate. As mentioned earlier, lower frame rates damage the immersiveness of the system, and distract the user from the interface.

One solution for this problem is using texture-mapping techniques. Texture mapping is the technique to cover a surface with an image. With this method, objects look more realistic, but their rendering is less computationally expensive. Unfortunately, the graphical libraries that were used for the prototype system could not handle texture mapping nicely. However, this technique was used successfully in the WWW-based system (Chapter 5). Figure 3.5 shows some of the symbols designed for the immersive system.

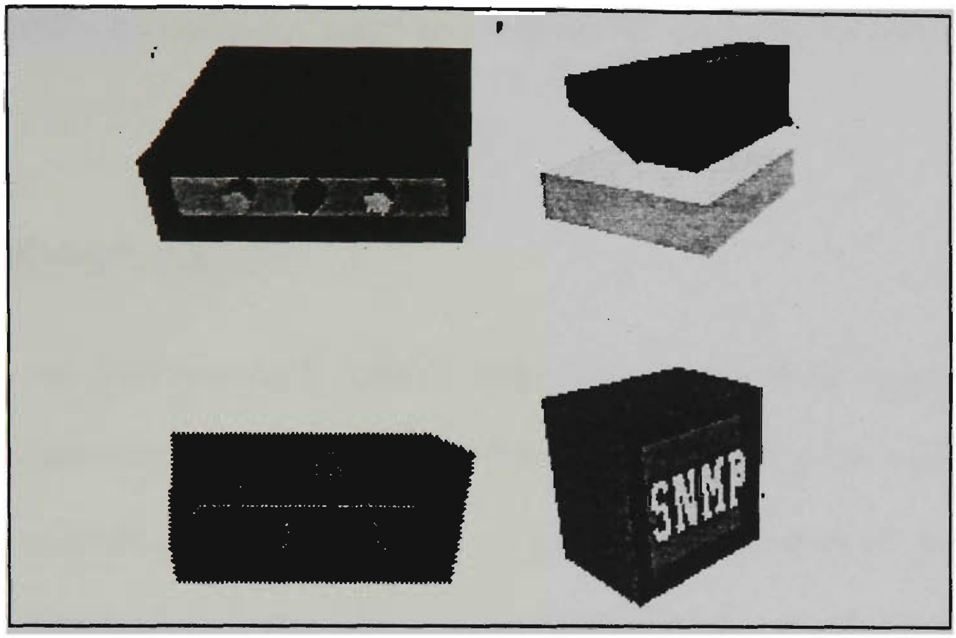

Figure 3.5- Some of 3D symbols representing network elements

To use the system, the user should firstly change the contents of the configuration file, so that it reflects the input and output devices that are available. Then, the system should be executed. The initial view is either the position of the viewpoint when the user left the system, or the top level of network view (based on command line arguments). The user can then, navigate through the virtual world using the input devices. As the result of navigation, the viewpoint changes and the change is reflected on the output device.

Objects in the virtual world are active so more information about their status can be obtained by walking into them for a detailed internal view. If the object is a network element, walking in will show the interfaces contained in the element. If the object is a sub-network, walking in will show the layout and status of the sub-network elements. In an ATM network, walking into a connection will show the virtual paths within the link. The walk in metaphor captures the hierarchically structure of the network and constrains the information presented on the display to potentially a comfortable level for network operators.

3.6 Observation

Based on the architecture and implementation issues, a prototype system was built. Using this prototype system, the user can observe the hierarchy of the network and its spatial relationship. The network can freely and quickly be navigated to observe the primitive information for network elements such as faulty devices and overloaded links (not in real-time). This can be achieved without becoming lost in a screen full of windows, the typical problem with existing WIMP based systems. Typical views of the prototype system are shown in Figure 3.6, Figure 3.7 and Figure 3.8.

Some basic manipulations of the objects are supported in this prototype. The user can use Edit mode and move objects around the scene, or change some attributes of the objects. These actions are included to show how efficient object manipulation can be done in an immersive virtual environment.

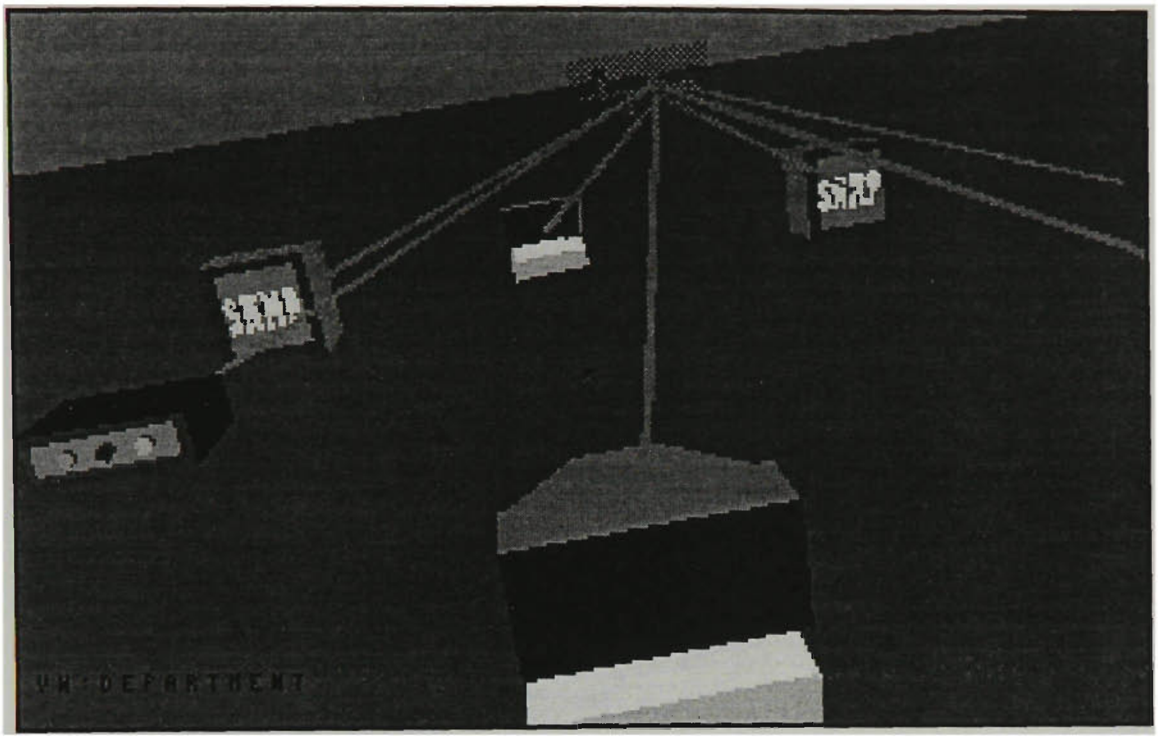

Figure 3.6- A view of the prototype system.

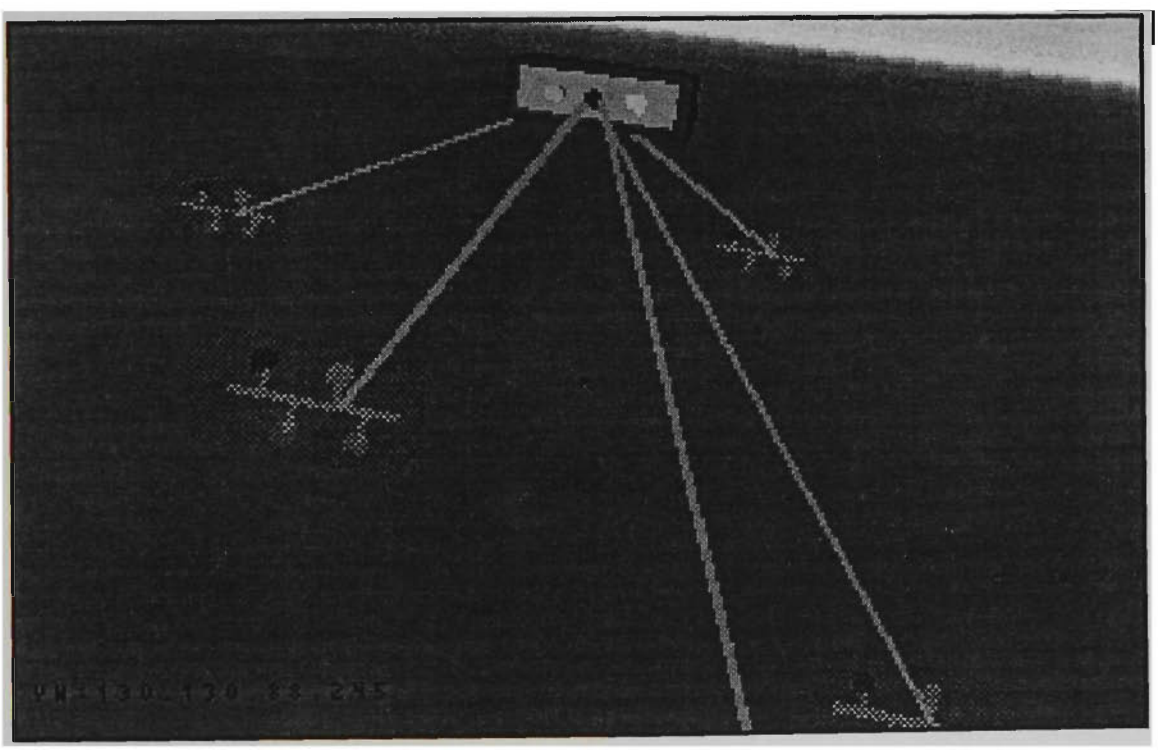

Figure 3.7- Another view of the prototype system (after navigating into gateway element of Figure 3.6)

The navigation and interaction can be done using any combination of input and output devices. For example, one could use the joystick and monitor, while another user can use HMD and Glove for this purpose.

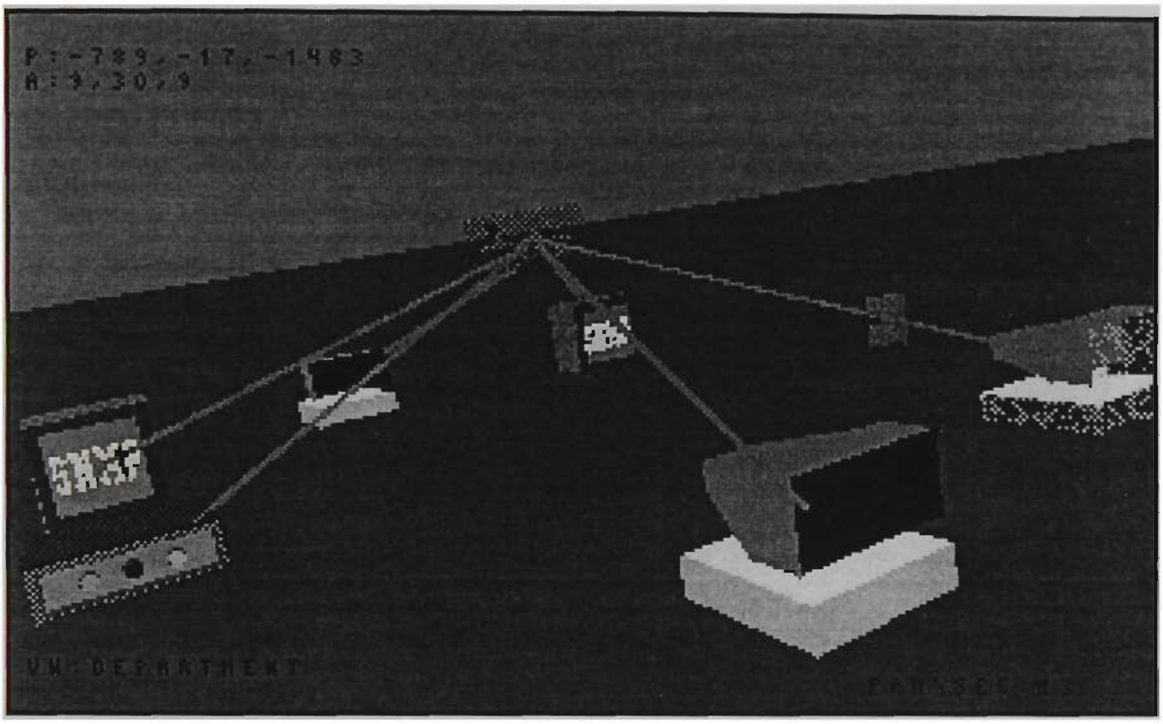

Figure 3.8- *Use of colour coding to show faulty devices and congested links.*

By doing several tests using a different combination of devices, it was found that if using HMD, working with a joystick and a mouse was rather difficult. However, using a joystick provided a better performance than a mouse in most circumstances. Using HMD and a glove provided higher immersion and more efficiency (in terms of navigation and interaction) than any other combination. Figure 3.9 shows a user with HMD and Glove interacting with the system.

Figure 3.6 and Figure 3.7 provide an example of how navigation is carried out. The user navigates into the gateway object at the left side of Figure 3.6. This takes the user into another view which is shown in Figure 3.7. This figure shows the subnets that are connected together via the gateway. Again, by navigating into any of these LAN symbols, the network elements within them can be examined.

One problem that was experienced using portals was that non-experienced users were immediately returned to the world that they have just exited. This problem has also been reported by other VR systems [Berger 94]. The problem was solved by moving the users' position a bit further into the new world, as soon as they enter it.

Figure 3.9- A user of the system wearing HMD and Glove.

In this prototype system, alarms and performance data were shown using colour coding as shown in Figure 3.8. Localised 3D sound, to locate alarms in the 3D world, where the devices were located; and speech recognition, for navigation and user commands, were considered but not implemented. However, speech commands were implemented in a student project and shown to be practical for navigation.

3.7 Qualitative Evaluation

After implementing the system, it was used by several people and an initial informal evaluation was conducted. The major finding of this evaluation is presented below.

One of the major advantages of the system, like any other well-designed VR user interface, is the short learning time. Using a HMD and joystick even a high school student quickly learnt how to navigate the system. This is because the interaction of the user with the system is designed to be as natural as possible. Therefore, there is little need to teach operators how to use the system. That is, if operators learn the basic principals of the interface, they can easily and quickly decide, when faced with more complicated situations, how to complete the task.

For instance, there is no need to teach operators how to move an object, because everybody knows how to move objects manually. This is in contrast to the WIMP user interface, in which all actions must be taught to the operator.

This advantage has a drawback for more complex tasks, however. As people's understandings varies, one person's representation may be someone else's misrepresentation [Stanger, 1992]. Therefore, VR user interfaces have to be flexible enough, so that each user could configure it so that it represents his own understanding. However, this feature was not incorporated into the prototype system.

Another advantage of this system is the extra spatial dimensions of the display. A three-dimensional VR display can accommodate more objects within a view without the user being lost or frustrated. In a two-dimensional display, if the length of view is larger than that of the window containing it, the user has to scroll to see all objects. Using an informal experiment, it was found that if the length of the view is three times more than the length of window, scrolling is cumbersome. To overcome this problem, the size of each object has to be minimised. However, a small icon does not leave enough space for showing necessary information for the corresponding network element.

Despite the advantages, the immersive VR user interface proved to have some serious drawbacks. As network management is nearly a continuous task, which takes several hours a day, the use of HMD causes some problems. Even the best available HMDs cause dizziness and eyestrain if worn for a long period. More serious symptoms have also been reported using immersive systems for a long period [Stanney, 1997]. HMDs have other minor problems as well. As the Inter-Pupillary Distances (IPD) of people varies, adjusting the HMD maybe required when another person wants to use it. More importantly, HMD obscures the user's view, which significantly reduces the interaction and communication of the operator in the real world.

The other problem is textual information. VR user interfaces can be designed so that they minimise the amount of textual data by representing them as symbols and sounds in the virtual world. However, in a network management environment there is a significant amount of information, such as MIB variables, that have to be presented to the operator as text. While small amounts of texts, such as the type of alarm, are quite manageable, in a 3D graphics-based user interface, the proper provision of lengthy texts is difficult and requires more investigation or better technology. In an immersive system with head mounted display, suddenly showing a view full of text significantly damages the feeling of engagement, and might even cause confusion for the user.

The other problem with immersive systems is their platform dependency. Most 3D input and output devices, such as HMDs and Gloves are available only for one or two platforms. That means, the users cannot use their existing high-end graphical workstations. Also, using bulky equipment reduces the mobility of the system, as it takes more time for all devices to be removed and installed in another location.

3.8 Conclusion

In this chapter, the design and implementation issues of an immersive user interface for network management were discussed. Firstly, the motivations for design of such a user interface were explained. Then, the architecture of the system was discussed.

An object-oriented approach was used for the design and implementation of the system, consisting of a set of classes and their relationships. This method resulted in a modular, maintainable and extensible system.

The visualisation method discussed in this chapter and in the paper [Kahani, 1995], was independently verified and used by a commercial network management system [Computer Associates TNG, 1996].

In this chapter, issues related to the implementation of the system were also discussed. This was followed by the observation of the system by providing a set of screen shots and appropriate descriptions.

It was noted that although the immersive VR had some advantages over traditional 2D user interfaces, it suffers from some drawbacks. In conclusion, it seems that the technology required for a full-feature immersive virtual reality user interface is not yet available.

Considering the limitations mentioned, it was decided to move toward a non-immersive approach, while maintaining the three-dimensional semantics of the view. Because of the need for a distributed and collaborative environment for effective management of forthcoming networks, a World Wide Web (WWW) based approach was chosen.

In the next chapter, the requirements for a distributed collaborative user interface are discussed. A generic protocol, GPMI, is developed to address the collaboration issues in these user interfaces. This protocol is used to build a WWW-based distributed collaborative user interface for network management (WNMS). This system will be discussed in Chapter 5.

Chapter 4

Multiuser Interface Design Issues

4.1 Introduction

The rapid growth of the Internet during recent years has paved the way for further implementation of network-dependent applications. These applications usually require a high level of connectivity so that users can access information from geographically dispersed locations. Networked games, Distributed Virtual Reality (DVR) systems, military simulation systems and Multiuser Interface (MUI) systems are among some systems using the network. In this chapter, issues related to 3D VR based multiuser interfaces are discussed.

Unlike other DVR applications, multiuser interface systems are persistent. A persistent system is one that exists and evolves even if there are no current users of the system. Common examples of persistent systems are network management systems and process control systems. Most persistent systems are event-based and log events to a data repository. The events modify the state of the system model in some way to reflect the real-world system being controlled.

Multiuser interface systems have another fundamental difference from other DVR applications. In most DVR systems, the most important objects are avatars (symbols representing participants). The semi-static objects, such as building and bridges, which are referred to as terrain, do not have a significant role in the overall system. Therefore, less attention has been paid to the problems related to them.

In a multiuser environment, the situation is reversed. That is, the most important elements are semi-static objects that are manipulated by users. The users, or their avatars, are not of much interest. The only important information about a user is the location and orientation of his avatar in the virtual world. This information determines the user's area of interest, so that only relevant information is presented to it. Consequently, MUIs are *object-centred,* as opposed to *avatar-centred* as in most DVR systems, and require the system to be optimised for this purpose.

In MUI systems the users' manipulations of the objects in the virtual world affect the functionality of a real world system. Therefore, MUIs usually have less tolerance for inconsistency. These differences question the suitability of the protocols defined for other DVR systems, such as DIS [IEEE-DIS, 1994], for multiuser contexts.

The issues addressed in this chapter and the developed protocol initially was considered and developed as part of an investigation to provide a collaborative VR-based user interface for management of telecommunication networks that is discussed in Chapter 5. The initial protocol was expanded later, and generalised to be a Generic Protocol for Multiuser Interfaces (GPMI).

Having a general protocol helps the user and developers in several ways. Firstly, most code can only be written once, but used in all other applications. This reusability of codes permits quick development of applications, as most of the program is already tested and can be used with more confidence. Moreover, using a general protocol increases the interoperability of different applications, potentially creating an integrated environment.

Initially, in this chapter, the proposed architecture for multiuser interfaces is explained. Then the GPMI protocol and its PDUs (Protocol Data Units) are discussed. To show the effectiveness of the system, the bandwidth requirement for a typical application is presented. The manner in which this protocol can be used in a multiuser interface is discussed in the next chapter by developing a multiuser interface for network management.

4.2 Architecture of VR Multiuser Interfaces

A virtual reality multiuser interface is a persistent 3D environment, in which several users collaboratively control a system, potentially in real-time. Similar to any other networked applications, it should provide a method for exchange of information among users, and between users and the actual real-world system. The user interface feature of the system enforces some concurrency control mechanism to ensure consistency among users. In addition, the persistency of the system requires special consideration. These issues are discussed in this section to establish the architecture of multiuser interface systems. This architecture provides a context in which the GPMI protocol can be used. The actual protocol definition is presented in the next section.

4.2.1 Communication method

The communication method is the way update data is exchanged among the participants. Communications can be categorised based on different aspects. In terms of the network layer, communications can be done via broadcasting, multicasting or unicasting. Categorisation can also be done with respect to the architecture. That is, systems can have distributed, hierarchical or client-server architectures.

In broadcast communications, the data is simply sent to every site in a local area network (LAN), regardless of whether the sites need, want or should receive it. In addition, the domain of communications is limited to a particular LAN. Broadcasting is useful for a distributed architecture with a dedicated LAN, as in SIMNET.

In multicasting, data is sent to one or several multicast addresses, and whoever wants to receive the data joins the appropriate multicast group. This method increases the efficiency of communications. Moreover, it does not restrict the system to a single LAN. As multicasting is not yet fully implemented on the Internet, islands of multicast-capable networks are connected together using a technique called 'tunnelling', which builds a multicast backbone (MBONE) over the Internet. This method has been used in a newer version of SIMNET, called NPSNET [Macedonia, 1995a]. DIVE also uses multicasting with negative acknowledgment (nack) for transmission of update information [Hagsand, 1996]. Multicasting is suitable for distributed systems as well as server-to-client communications in a client-server context.

For point-to-point communications, or unicast, one connection is established for each participant. In this model, either each user connects to all other users, or all users connect to a server (client-server). For the former, some limiting factors are required to restrict the number of connections. Otherwise, the number of connections grows too rapidly with respect to the number of users ($O(n^2)$). This method has been used in the MASSIVE project [Greenhalgh, 1995].

In the client-server model all participants are connected to a central location (referred to as Collaboration Manager (CM) hereafter), data sent to the CM are resent to all other interested users. While unicasting is used to send data to the CM, unicasting or multicasting can be used for transmission of data to clients. This method has been used in the NVR system [Berger, 1994], and BrickNet [Singh, 1994].

The client-server architecture is more suitable for multiuser interface systems, for several reasons. As one of the important characteristics of multiuser interface systems is their persistency, a multiuser interface system should have a data repository. A copy of all exchanged data has to be sent to this database to keep it current. Therefore, it is already a good nominee to act as the CM. It is also easier to provide the required strict consistency among the user and the database in a client-server system than other architectures.

The client-server architecture has two main drawbacks that should be addressed. Firstly, the CM is a single point of failure. Secondly, with large numbers of users, the CM becomes a bottleneck in both networking and processing. The first problem would result in a system breakdown even in a distributed architecture, as the failure of the database collapses the whole system. That is, if the users' manipulations of virtual objects can not be reflected to the real world system, because of database breakdown, the whole operation becomes useless.

In most multiuser interface environments the number of users is not too high (eg. less than ten), as only a few people need, want, or should have access to the system, mostly because of the sensitivity of the information. However, in some systems, such as a nation-wide telecommunication system, the number could be many hundreds. For those systems, a distributed database should be used to reduce the bottlenecks.

Although the GPMI protocol does not address communication among distributed databases, there are some recommendations on how to divide the task among databases. In this regard, the main objective is to minimise not only the amounts of load for each server, but also the intersections and interactions among databases. If databases have fewer common points, there is less information exchanged among them, which reduces the latency.

For instance, controlling a big factory might require dividing the whole virtual world into a set of departments or sections. This breakdown reduces the number of users and objects, hence decreasing the load. As each virtual world is connected to others via portals, the amount of traffic among servers is also minimised. It is also possible to divide the database based on functionality, eg. electrical section versus mechanical section. With this method, although users see all sections integrated into one view, manipulating each section requires notification of the appropriate server. Using this breakdown, the number of update messages for each server is reduced; though, depending on the implementation, the amount of interaction between servers might be quite high.

The client-server architecture has some other benefits, as well. First of all, as communication with the data repository is only through one channel, the consistency control and concurrency control are much simpler. Secondly, each client needs to communicate only with one computer. This will allow the establishment of a connection and its use for all data exchange. Moreover, using a reliable transport layer, removes the need to resend update information repeatedly to all participants to overcome the problem of missing packets.

4.2.1.1 Area of Interest (AOI)

In a user interface with several independent, or semi-independent, tasks and many objects, groups of users simultaneously work on different parts of the system. At any moment only a limited number of users are interested in receiving update data for a particular object. Therefore, by defining an area of interest (AOI) for each user, and only forwarding relevant information to that user, a considerable amount of bandwidth can be saved.

The attributes of AOIs, including their shape, are completely application-dependent. For instance, in military simulation systems the definition and the shape of these areas are quite different from that of a network management application. In NPSNET

[Macedonia, 1995a], the AOIs are hexagonal cells, with each cell associated with a multicast group. Users have membership of several multicast groups depending on their location in the simulation. As they move, they leave some groups and join others.

In the latest version of DIVE [Hagsand, 1996], a light-weight grouping has been used to make the area of interest or aura of each user smaller. Even within a world, users tend to gather in several locations, and the user that is within a group, is not interested in receiving the update data related to other groups.

Spline (Scalable Platform for Large Interactive Networked Environments) [Barrus, 1996] uses another approach. It divides a virtual world into chunks, called locales, which can be processed separately. The division has been left as an implementation issue, and is transparent to the users. In Spline, each user only sees the containing locale and some neighbouring locales at once, and can only interact with them.

In WNMS (Web-based network management system) [Kahani, 1996] the AOI has been defined as the current level of hierarchy in the network that the user is within. For instance, if a user is navigating within a local area network, his AOI is all network elements and all other managers within that particular LAN segment and its neighbourhood. If a user oversees the network from the campus level, that user views each LAN as an object. That is, user only receives update data concerning the overall function of each LAN, and not all the data for all individual objects within LANs.

Client-server architecture enables the server to decide to whom data should be transmitted. Depending on the application, this task might be computationally expensive. However, the server can use multicasting to distribute this burden over network routers. That is, although the server receives data from clients via unicast TCP/IP connections, it can send the data via a reliable multicast protocol such as Multicast Transport Protocol (MTP) [Armstrong, 1992].

4.2.1.2 Object behaviour

Behaviour can be defined as the ever-changing state of the attributes of an object or entity. Entities can be divided into two major categories: those whose behaviours are deterministic and those whose behaviours are non-deterministic, each of these groups is further divided into two groups [Roehl, 1995b]. Entities with deterministic behaviour are categorised further into static and animated entities. Static entities, such as mountains or buildings, never change during the simulation. Animated entities change over time, but their states are easily predictable. An example would be the movement of a clock's hands.

Non-deterministic entities consist of Newtonian and intelligent entities. Newtonian entities respond to stimuli in their environment according to the laws that their creator implemented. Intelligent entities, such as human beings, have a complex behaviour, which cannot be easily predicted.

Considering this taxonomy, four levels of behaviour, corresponding to four types of entities, can be distinguished. In level 0, attributes are modified directly, such as "move object A to location x, y, z". In level 1, attributes change over time, such as "move object A at speed of x in direction of y". In level 2, a series of calls to level 1 behaviours are performed, such as "adjust objects to a particular format". In level 3, high-level decisions are made, such as "decide whether to isolate an element, use backup device, etc." in a network management environment. It has been suggested the network interface should be placed between level 1 and 2 [Roehl, 1995b]. Putting interfacing between level 0 and 1, while simplifying the client's system, places a great burden over the network. Interfacing between level 2 and 3 requires all simulators to be absolutely up to date at every frame, which is difficult to maintain. The implementation also becomes very complicated.

The GPMI protocol provides facilities to place the network interface both between level 0 and 1, and between level 1 and 2. However, changing attributes of objects might cause the system to perform higher level activities. For example, changing the status of a network element might cause a change in its colour, blinking and generation of a sound alarm.

4.2.2 Consistency control

Consistency control is one of the important issues in a collaborative environment. However, due to the complexity of efficient consistency and concurrency controls, most CSCW systems only use social protocols [Beadle, 1995; Roehl, 1995b]. However, this method is not suitable for multiuser interfaces, particularly in network management contexts, as they require more restricted consistency. In general, there are two possible ways to control the consistency in these environments: explicit locking and implicit locking.

Explicit locking means that the user explicitly locks the pieces that he wants to work on, and explicitly releases them when the task is finished [Roehl, 1997]. This task is done in most DVR systems by letting users or processes own an object [Kessler, 1996]. In the client-server environment, the server locks objects upon the request of the clients. Explicit locking is appropriate if a user wants to manipulate a set of objects, and does not want anyone interfering until the whole task is finished.

Explicit locking is not as straightforward as it looks. The problem arises when users leave or are cut off in the virtual world without releasing the objects they have locked. In most DVR systems the main issue is detecting those orphan locks and releasing them. This problem can be easily addressed in a multiuser environment, by letting the CM keep a record of objects locked by each user. As soon as a user leaves the system, the CM realises that (because of the connection termination message), and releases the locked objects.

However, in a user interface context, users might require that objects remain locked even after they leave the system. This allows a user to fulfil a time-consuming task without interference from other users. Depending on the type of application, this task can be handled differently. The simplest way of handling this feature is for the client program to ask whether the user would like to release the locks, and upon affirmation release them explicitly.

Unlike explicit locking, implicit locking requires no lock/release messages. The required locking is done automatically by the CM by letting each object record the last user that has manipulated it and the last access time. Therefore, whenever the CM receives an access request for an object, it checks the attributes of the object. If the same user wants to manipulate the object, the access is granted and the last access time field is updated. If another user requires access, the CM checks the time of the last access. If the elapsed time is bigger than a specific interval, it is assumed that the previous user has finished with the object, and access is granted. Otherwise, access is denied and the transaction is rejected.

Explicit locking provides more restricted consistency than implicit locking, as it requires lock acceptance before any manipulation of the objects. Implicit locking might increase the level of collaboration by eliminating the need for lock and release messages. It is simpler to implement at the client side if roll-back is possible. However, in those systems where roll-back is impossible or is expensive, the use of implicit locking is not recommended.

For both methods, it might be more suitable to lock several interdependent attributes of an object, rather than locking the whole object. This allows manipulation of different attributes of an object by several simultaneous users. This feature is provided by the GPMI protocol.

4.2.3 Database

The type of database plays a significant role in the overall system. Some systems, such as SIMNET, have been based on the assumption that each user has a fully replicated copy of the database before entering the simulation. All modifications to the original database are continuously broadcasted, so that newcomers can easily update their database.

This assumption cannot be fulfilled in all multiuser interface systems. First, as the database is continuously changing, it may not possible to have a fully replicated database before entering the virtual environment. Therefore, upon entering the system, each participant should receive the most updated version of the database either completely, or partially. Full replication requires a larger memory at the client side, and is therefore difficult to keep up to date. Consequently, most DVR systems [Hagsand, 1996; Roehl, 1995a] use a partially replicated database model.

4.3 Protocol Definitions

The Generic Protocol for Multiuser Interfaces (GPMI) has been designed to support the virtual reality multiuser interface architecture discussed in the previous section. The protocol provides the facility to build strictly consistent environments and supports the different functionality required in these contexts. The structure of the GPMI protocol is as follows:

- GPMI supports the client-server architecture. The clients establish a unicast connection to the server, through which they send update information to the server. The server, however, can either use that connection, or use reliable multicasting to send update information to the clients.

- GPMI supports a partially replicated database model. Clients receive the part of the database that they are interested in as they connect to the system. The startup data can be transmitted via either GPMI or another protocol, eg. HTTP. As the users' area of interest changes, they receive the relevant section of the database.

- The server determines the area of interest of each user.

- Each client sends its position and orientation repeatedly (every five seconds or so). This signal acts as a heartbeat, and allows the server to quickly realise if any client is disconnected.

- GPMI support both explicit and implicit locking. Although in explicit locking mechanism lock and release messages have to be exchanged between server and clients, no message is required for implicit locking.

- GPMI supports both level 0 and 1 of behaviour. That is, it is possible to send the position of an object as it changes. The velocity of an object in a specific direction can also be transmitted, allowing dead reckoning technique [DIS, 1994] be used to determine the position of the object at a later time. It also should be noted that unlike DIS, dead reckoning in GPMI is not restricted to the position and orientation, and can be used for any attributes of the objects.

- GPMI recommends thresholding [Berger, 1994] for transmission of level 0 behaviours. Thresholding means that clients should broadcast their update notifications every *n*th frame rather that every frame. This is because a powerful computer can provide a relatively high frame rate (50 or higher). If changes to the level 0 behaviours are submitted every frame, it generates too much traffic and might flood slower computers. The actual value of *n* depends on the application and the technology.

- To lower the bandwidth requirement and allow the use of the system from bandwidth-scarce connections, the clients should have the option of not receiving the update message related to the avatars of other users. With this option, only modifications to the objects are sent to the user. The information related to the movement of the user, however, has to be submitted to allow the server to determine the area of interest of the user.

4.3.1 Data type definition

The data types in GPMI are based on eXternal Data Representation (XDR) standard [Srinivasan, 1995]. XDR is an Internet standard for the description and encoding of data. It is useful for transferring data between different computer architectures. It is similar to the ISO X.409 Standard with the major difference between them being that XDR uses implicit typing, while X.409 uses explicit typing.

XDR allows definition of fixed-length and variable-length user defined types (called opaque), and supports structures, unions, arrays, etc. This feature provides flexibility for the system to send a set of attributes of an object in one message. For instance, it is possible to define a structure consisting of position and orientation attributes and send together or individually

4.3.2 PDU definition

Currently, only five types of messages or Protocol data Units (PDUs) for GPMI protocol have been defined. Although these PDUs are general enough to support most applications, new PDUs may be added in the future to support new features or to specialise the current ones. The definitions and applications of these five PDUs will be discussed in the following subsections. The PDUs are:

1. Update Notification PDU (UNPDU)
2. Update Notification with Dead Reckoning PDU (UNDRPDU)
3. Object PDU (OPDU)
4. Lock/Release PDU (LRPDU)
5. Message PDU (MPDU)

4.3.2.1 Update Notification PDU (UNPDU)

The UNPDU is issued by a client program or the CM to report the change in an attribute of an object. The PDU is used if the attribute does not use the dead reckoning technique. For example, to report that the description of an object has changed, a UNPDU has to be issued by the client that has caused the change.

All types of PDUs have a field that shows the type of the PDU. That is, whether the PDU is a normal PDU, an acknowledge PDU, or a rejected PDU. All PDUs initiated by clients or the CM are tagged as normal. The receiver sends an acknowledgement back to the sender. Each PDU also has a field called the *pduSequence*. This field contains a sequence number given to each generated PDU, and is incremented locally. The acknowledged message in conjunction with the *pduSequence* field can be used as a positive acknowledgment.

As the underlying communication protocol is reliable, it guarantees the receiving and the order of the messages in most situations. However, if the connection is broken due to a network failure, there might be some inconsistency between sender and receiver. That is, there are some messages that may have been received, but as the TCP *ack* signal could not reach the sender, the sender assumes those message have not been delivered. Depending on the implementation, if the receiver can deal with these situations, there is no need for using acknowledgment. If the sensitivity of data is not high, or the probability of network failure is low, the acknowledgment for UNPDUs can be omitted to save some bandwidth and reduce latency. In this case, as the client reconnects, it should receive the most recent version of the part of the world in its AOI from the server to synchronise its view with others.

The rejected PDU act as an negative acknowledgment and its usage is mandatory. It is used mostly for implicit locking, as in explicit locking objects are firstly locked before manipulation. It can be used to apply some restrictions on users' activities. For example, the rejected PDU can be used to limit moving an object beyond a defined boundary if it is not possible to do that through the client's software.

Note that on receiving a rejected PDU, the client should undo the operation that has been rejected to keep its local view consistent with other users views. Therefore, the required information for undo operation should be included within the rejected PDU. For example, if a UNPDU has been rejected, the rejected PDU should contain the latest value of the attributes of the object it wanted to change, so that the receiver can update its view.

In addition, a piggyback scheme has been considered for all PDUs. Whenever, a client or the CM wants to send a PDU, it sends the *pduSequence* of the last received PDU in the *piggyback* field to indicate a positive acknowledgment. Although piggyback can save some bandwidth, the parties should not delay sending the acknowledge message until they have a PDU to submit, so that it affects the real-time feature of the application.

4.3.2.2 Update Notification with Dead Reckoning PDU (UNDRPDU)

The UNDRPDU is also issued by a client program or CM to report a change in one attribute of an object. However, this PDU is issued for those attributes that use the dead reckoning algorithm [IEEE DIS, 1994], such as position and orientation. Although it reduces the volume of traffic, using the dead reckoning algorithm places an extra processing burden on both sender and receiver sides. Therefore, it should only be used when the extra cost is justified by the amount of traffic it saves.

The justification should be made at the design time by considering the characteristics of the attributes of each object. Those attributes that change continuously, such as position and orientation, are good candidates for applying dead reckoning. Then, the manner in which the attribute changes should be examined. For instance, the position of a vehicle is quite suitable for dead reckoning, as it does not change direction too often. The position of a human in the virtual world is not suitable for dead reckoning, as it has a level 3 behaviour and changes direction often, hence, changes cannot be easily predicted. Note that for any significant change in the direction or speed, a new PDU should be transmitted.

It should be clarified that XDR data type allows defining a structure consisting of position and velocity vector. Therefore, it is possible to use UNPDU to transmit dead reckoning parameters. However, defining another PDU might simplify the parsing, especially when applied to simple and single attributes.

4.3.2.3 Object PDU (OPDU)

An OPDU is issued to notify creation or deletion of an object or an avatar. Whenever a user enters a virtual world, its avatar should be created in the other users' views, so that they can see the arrival of the new user. The avatar has to be deleted when the user leaves the virtual world as well. Moreover, although the objects within each virtual world are sent to the clients as they enter a world, the creation of new objects or deletion of the existing ones should be reported to them, as well. The OPDU provides a general format that allows all of these operations.

All OPDU messages require positive or negative acknowledgments depending on their acceptance or rejection by the receiver. The program implementing this PDU should be able to undo the operation if the PDU is rejected.

It should be noted that the GPMI protocol does not provide any special format for avatars and objects. Therefore, the application that implements the protocol should decide to either define a proprietary format or uses the *de facto* standard avatar formats, such as Universal Avatar [Ma, 1996] or Living World [LW Specification, 1997].

4.3.2.4 Lock/Release PDU (LRPDU)

A LRPDU is issued to indicate that a user has locked an object to change its attribute, or has released it. This PDU is only used for explicit locking mechanism. Implicit locking mechanism should be provided by the CM, assuming that all modifications to the objects are locked. The LRPDU provides facility to lock an attribute of the object. This allows several users to modify different attributes of an object, concurrently.

Implicit and explicit locking mechanisms are very similar, with the exception that implicit locking lacks lock and release PDUs. Instead locking is done with the first interaction of a user with an object, and releasing is done after a period of no interaction with the object. Figure 4.1 illustrates the explicit locking mechanism in a two-user environment. Firstly, Object O_i is locked by User 1, so the access of User 2 is rejected. After release of the object by User 1, User 2 can access, and manipulate it.

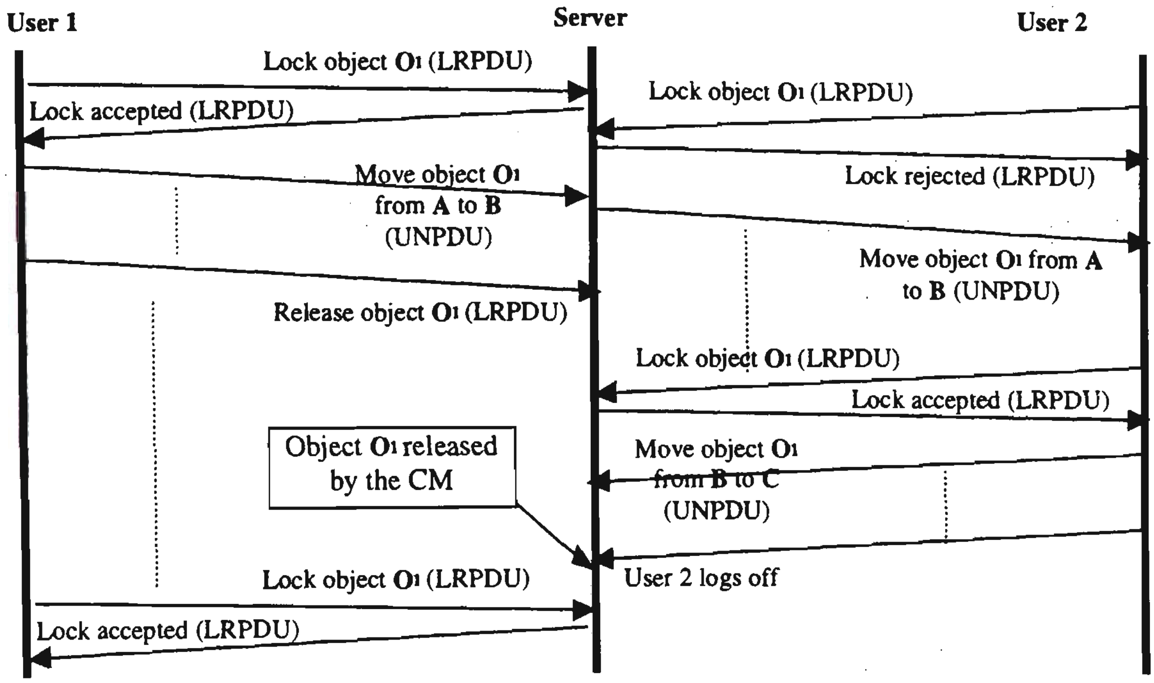

Figure 4.1- Explicit locking mechanism (only relevant messages are shown)

Figure 4.2 shows how implicit locking can be implemented. User 2 tries to move O_1. Later on, User 1 tries to access the same object, but the access is rejected, as the object is already locked by another user. As the rejected PDU has the latest position of Object O_1, it is possible for User 1 to roll-back its operation and be consistent with other users. After a while, User 2 tries to access the same object. As the lock has been expired by that time, the access is granted, and User 1's subsequent access attempt is rejected. Note that, if roll-back operation for the rejected PDUs is not possible, implicit locking should not be used. Also, the amount of time-out is application-dependent and should be chosen practically.

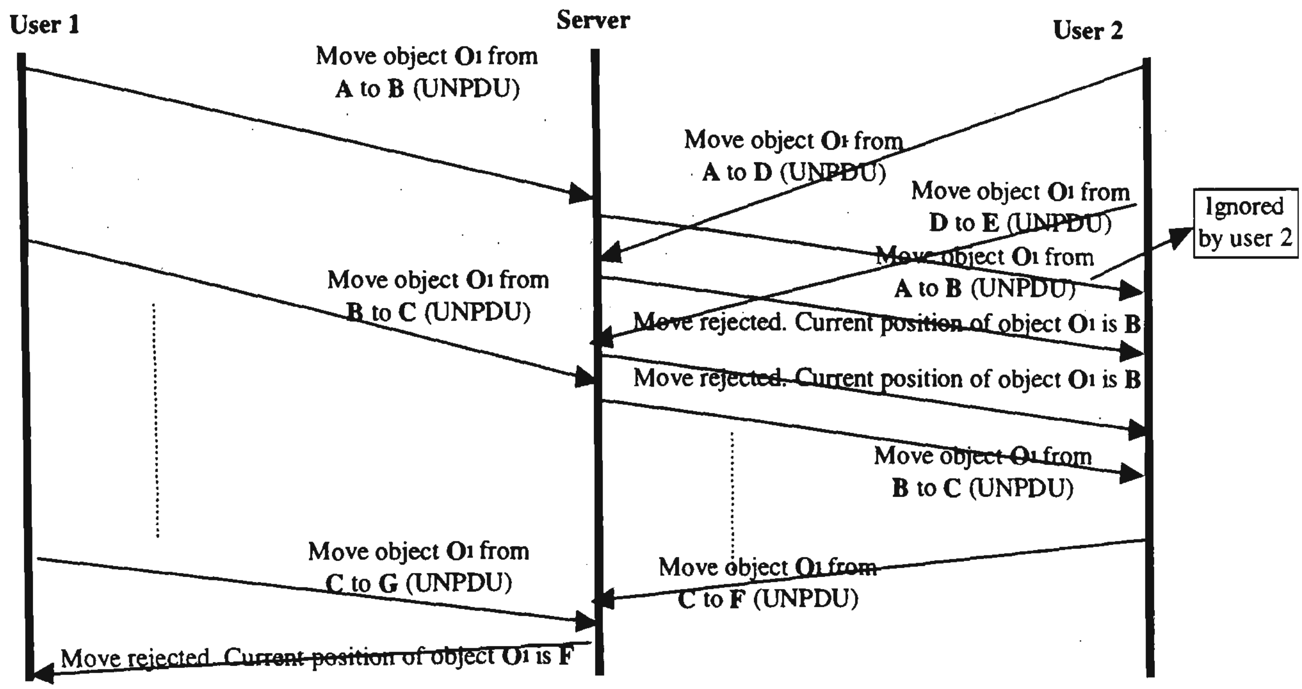

Figure 4.2- Implicit locking mechanism (only relevant messages are shown)

Similar to the OPDU, all LRPDU messages need to be acknowledged. If the lock is rejected, the program should notify the user in some way. For example, if the object has been highlighted, it should be changed to normal.

4.3.2.5 Message PDU (MPDU)

An MPDU is a general message and is issued when a user requires a special service from the CM, or wants to send a message to another user in the system. For example, a user might use this PDU to notify the CM that it has passed the initialisation phase and is ready to receive update notifications. In addition, it might use this PDU to send a message to other users. The different uses for this PDU are application dependent. Upon receiving a MPDU, the receiver should acknowledge it either explicitly or via the piggyback scheme.

Using an MPDU, each client can send information to any other user. So, this PDU can easily be used to implement a textual chat interface between users. Although it is possible to use the MPDU for some real-time non-textual data (eg. voice), the synchronisation of this data might be difficult. Instead, users can use MPDU services to establish a direct link between themselves for transmission of real-time information.

4.4 Bandwidth Requirements

It is desirable to estimate bandwidth required for each user, and the total bandwidth requirement for the CM. However, as this data is application-dependent, it is difficult to provide an accurate estimate. Therefore, it was decided to use an example and by measuring the traffic for that system, provide some estimate value. For this purpose, the Web-based Network Management System (WNMS) [Kahani, 1996] was selected. More detailed information about this system is presented in Chapter 5.

In addition, as the GPMI protocol only addresses collaboration among the parties, initial download of the virtual world is not of interest, here. To provide some value, the amount of traffic exchanged between the CM and a client in WNMS was measured.. To consider the worst situation, the user was continuously active during the measurement period. This is not a likely situation in everyday usage where at most of the time users are watching rather than acting. Nevertheless, at the time of a network failure, in which collaboration is useful, the users are mostly active. It should be noted that for traffic modeling, the system should be used in a real environment, and the traffic monitored for a very long period. It should also be considered that the characteristics of this traffic would be application-dependent.

The measured traffic had an average of 250 bytes/sec and a peak of 650 bytes/sec. However, this traffic will be higher if the frame rate increases (using more powerful computers). Because the VRML browser used for the experiment was not optimised, it could not support high frame rate even for a mediumly populated virtual world (eg. with 15-20 network elements). The use of the EAI (External Application Interface) for networking further decreased the frame rate. Although in immersive VR environments, frame rates of 15 or higher are required, for desktop VR systems, a frame rate of 10 proved to be satisfactory, experimentally. Considering the frame rate factor, it would be expected that the average and peak rate of collaboration traffic are approximately 1K/sec and 2K/sec, respectively.

Although this is a negligible amount of bandwidth, it should be considered that as the number of users increases, the amount of traffic for the CM and for each user increases. This is because modifications done by each user have to be sent to the other users. Depending on whether users belong to one AOI or several AOIs, and the activeness or passiveness of the users, the bandwidth usage varies, dramatically.

Table 4.1 shows the effect of the number of users and their positions in the virtual world (AOI) under different situations. The first row of the table, assumes that N users are fully active and within a single AOI. Each user generates on average **B** bytes of traffic. Therefore, each user sends **B** bytes of traffic to the CM, and receives **(N-1)×B** from it.

This causes (**N** × **B**) bytes of traffic for each connection and (N^2 × **B**) bytes for the CM, overally.

The second row of the table, illustrates the effect of the AOI. Assuming that there are **M** AOIs and **N** users are evenly distributed among them, each AOI would includes (**N/M**) users. In this case, the overall traffic is reduced by a factor of **M**. However, in a real environment, not all the users are active, simultaneously. This factor has been considered in the third row of Table 4.1. With **P** users active in each AOI, the total traffic for the CM is decreased to (**P** × **N** × **B**), and for each user to (**P** × **B**).

Situation	Total traffic at the CM		Total traffic for each user	
	Formula	eg.: N=30, B=1Kbyte/s	Formula	eg.: N=30, B=1Kbyte/s
N users are active and in one AOI	$N^2 \times B$	**900 K**	**N×B**	**30 K**
N users are active but distributed evenly in **M** AOI	$N^2/M \times B$	**M = 10 ⇒ 90 K** **M = 5 ⇒ 180 K**	**N/M×B**	**M = 10 ⇒ 3 K** **M = 5 ⇒ 6 K**
N users distributed evenly in **M** AOI, only **P** users in each AOI are active simultaneously	**P×N×B**	**M = 10, P = 1 ⇒30 K** **M = 5, P = 2 ⇒ 60 K** **M = 1, P = 10 ⇒ 300K**	**P×B**	**M = 10, P = 1 ⇒1 K** **M = 5, P = 2 ⇒ 2 K** **M = 1, P = 10 ⇒10 K**

Table 4.1- Collaboration traffic of WNMS under different situations.

As shown in Table 4.1 the Area Of Interest (AOI) can result in a reduction of bandwidth requirement. However, it should be considered that it increases the processing requirement at the CM, as the CM should decide where a packet should be sent. Therefore, depending on the application, the consideration of AOI might be computationally expensive. The excessive processing requirement will add more latency to the packets, which is not desirable.

4.5 Conclusion

In this chapter, the requirements and architecture of a multiuser interface system were described and the development of GPMI, a generic protocol for virtual reality multiuser interfaces, was discussed. The requirements were categorised as communication, consistency and concurrency control, and database issues. The most important issues in each category were discussed, and suitable solutions for multiuser interface environments were presented.

Based upon the discussed requirements, the GPMI protocol was defined. GPMI consists of five PDUs, and is general enough to support most multiuser systems. Following the explanation of protocol details, bandwidth requirements were discussed. Measuring the traffic and analysing it in several different circumstances revealed that the amount of traffic for a medium number of users is quite manageable. However, it was mentioned that the processing requirements, though application-dependent, might be high.

In the next chapter, the protocol developed in this chapter will be used to build a collaborative distributed multiuser interface for network management. It will be shown how and under what circumstances each PDU of the protocol should be used.

Chapter 5

WWW-based Collaborative User Interface

5.1 Introduction

In Chapter 3, the architecture of an immersive VR user interface for network management was discussed. After human studies and qualitative evaluation, it was concluded that the required technology for an immersive system is not yet available. In this chapter, a desktop VR user interface for network management is described.

The user interface is based on World Wide Web (WWW) browsers using both text (HTML) and 3D graphics (VRML). This means that the interface is cheap, ubiquitous and platform-independent.

As mentioned in Chapter 1, the new technology in broadband networks provides much more flexibility and configurability than was previously available. For example, ATM networks facilitate the implementation of logical connectivity and virtual private networks [Kositpaiboon, 1993]. This feature allows several logically separate networks to share the same physical network. As failure of the physical link affects all involved parties, they should have been given a role in network management. Moreover, as prescribed by [ATM Forum-CNM, 1996], customers need to negotiate with their service providers to change their requirements in real-time.

To further investigate this issue, collaboration was added to the Web-based user interface. The protocol developed in Chapter 4 was used for this purpose. In this collaborative environment, not only different private network managers can coordinate their works with each other, but also different operators within a private network can collaborate with each other. The collaboration could also facilitate fault management.

In this chapter, firstly, some background information on WWW technologies and protocols are provided. Then, the special requirements of a distributed user interface for network management is discussed, followed by the explanation of the architecture of a Web-based desktop VR user interface. Some of the most important implementation issues, observations and a qualitative evaluation are then presented. The rigorous and quantitative comparison of this system (WNMS) with traditional user interfaces is discussed in Chapter 6.

5.2 World Wide Web (WWW) Technologies

The Internet has grown dramatically in recent years, and has become the main tool for interconnecting computers and systems. The emergence of the World Wide Web (WWW) and hypertext technology has transformed the Internet from mostly text-based email messages and bulk file transfer into multimedia applications consisting of voice and video clips. The WWW uses HyperText Transfer Protocol (HTTP), layered on top of TCP/IP, to transfer HyperText Markup Language (HTML) documents between computers. An HTML document consists of a set of anchors and links that relate it to other documents.

5.2.1 Virtual Reality Modelling Language (VRML)

Although the evolution of HTML was a major break-through, it is still based on text and a set of two dimensional images and voice clips. To further enhance it, and allow the use of three-dimensional graphics, Virtual Reality Modelling Language (VRML) was developed. VRML has been designed to be a "universal description language for multi-participant simulation" and to meet the following requirements: platform independency, extensibility, and the ability to work well over low-bandwidth connections [Bell, 1995].

VRML Version 1.0 does not support interactive behaviours. That is, the world is static and the user only can navigate it. However, Version 2.0 [Bell, 1996] supports object's behaviour and allows nodes to have input and output parameters. Moreover, it provides an external API (EAI), which can be used to link VRML to languages such as C++ or Java.

VRML is a hierarchical object-oriented script in which each object has a set of attributes. Attributes also are objects and can contain some other objects. The

VRML specification defines a set of important and essential nodes. However, using PROTO and EXTERNPROTO keywords, it is possible to define a customised node and use it later. Figure 5.1 shows sample VRML code that adds a box into the virtual world.

```
Transform {                     # wrap everything
   position 1, 5, -1            # position following object[s] at x=1, y = 5, z= -1
   rotation 1, 0, 0, 1.57       # rotate following object[s] 90° around X axis
   children [             # list of object[s]
      Shape {                   # a visible object
         appearance Appearance {
            material Material {
               diffuseColor 0 0 .5        # color is blue
            }
         }
         geometry Box{size .4 .5 .12}    # object is a box with the size
      }
   ]
}
```

Figure 5.1- VRML sample code.

5.2.2 Java Language

Java language is a class-based object-oriented language, which has been designed to be machine-independent and platform-independent [Sun Microsystem, 1996]. Java is syntactically similar to C++, however, it is conceptually different from C++ in several ways.

Unlike C++, Java is only compiled (not linked). This feature provides the unique characteristic of Java, which is platform independency. In addition, some complicated and error prone features of C++, such as multiple inheritance and templates, have been omitted in Java, to make it simpler, more efficient and easy to program. Finally, the automatic garbage collection of Java releases programmers from the annoying task of deleting unwanted variables.

```
public class mkQueue {
        // member variables declaration
        public Object [] queue;              //Queue itself
        public int queueLength;              // Length of the queue
        public int current, last;            // several required pointers
        public boolean overwrite;            // overwrite option

        //function definition
        public mkQueue (int ql) {            //Constructor
                // Initialise vaiables
                queueLength = ql;
                queue = new Object [ql];
                current = last = 0;
                overwrite=false;
        }
        //pushes a string into the queue; returns 0, if it is full and no overwrite
        public synchronized int push (Object buffer)
        {
        ......
        }

        // Pops a string from the queue; null if queue is empty
        public synchronized Object pop ()
        {
        ......
        }
        // other functions
        ....
}
```

Figure 5.2- A sample code written in Java.

Java can be used in two methods: standalone program or applet. Standalone programs run on a computer using a Java interpreter. Java applets are used by web-browsers and are combined with HTML or VRML files to enhance their capabilities. Figure 5.2 shows sample code in Java defining a class.

Using Java and VRML, it is possible to build a distributed multi-user three-dimensional environment in which a user can see the representative symbols of other participants (avatars), and collaborate with them in a virtual environment. This environment has been utilised to create a network management user interface in which operators can collaboratively manage potentially more complex networks.

5.3 System Requirement

The proposed system is a distributed virtual reality (DVR) system. DVR systems are mostly used for simulations, such as SIMNET [Pope, 1989], or computer games. In a network management environment, several issues have to be treated differently:

- **Bandwidth**. Although, currently, in most telecommunication systems, a separate network is used for network management traffic, the ability to use the data network to access the management information is desirable. This provides more accessibility and allows remote monitoring of the network. In this case, the traffic generated by the user interface uses the same bandwidth available for network elements. Therefore, the required bandwidth for the user interface should be as small as possible. That is, unlike some other systems in which the distributed system itself is the goal (eg. games or simulators), the user interface is a tool for achieving the goal (managing the network to provide a desired QoS). Therefore, the tool should not undermine the goal by using a significant amount of bandwidth, reducing the throughput and consequently QoS.

- **Reliability**. Reliability is a major issue in network management. In a distributed VR game if some update messages are lost, the effect on the overall system is not dramatic. In a network management environment, each message may carry important information, and it may have a catastrophic effects on the network if it does not reach the destination. To fulfil both reliability and bandwidth restriction conditions, a reliable end-to-end protocol, such as TCP/IP, instead of a best effort protocol, should be used.

- **Number of Users**. Unlike some distributed simulations that have a large number of participants, the number of users in a network management interface is low to medium. This is because only a few people are authorised to access the information. Although there might be a relatively large number of users in a countrywide telecommunication network management system, as explained in the next section, it can be broken into several smaller segments. Therefore, the protocol should support a medium number of users, eg. 30-50.

- **Security**. For most distributed simulation systems, security is not an issue. Some of these systems have a dedicated network, which physically maintains the security, for others, such as distributed games, the data is not sensitive.

None of these is true for a network management environment. Therefore, special security measures have to be considered.

These requirements meet the criteria discussed in Chapter 4 for collaborative multiuser interfaces. So, the GPMI protocol was developed and used to address the collaboration issues of the user interface.

5.4 System Architecture

The system uses a client/server architecture and is based on the Telecommunication Management Network (TMN) model [ITU-T Recommendation M.3010, 1992]. As shown in Figure 5.3, the system consists of a set of servers acting as proxies for network management systems (NMS). Each server communicates with a network management system and uses its services to get the management information. This information is then sent to the clients, which are WWW browsers enhanced with VRML, Java and JavaScript. Each VRML object can have a link to other views that may be within the domain of another NMS. This allows an integrated view of distributed networks in which each subnet is managed by an independent NMS. Managers can also collaborate with each other, in real-time, to solve the problems that involve more than one domain.

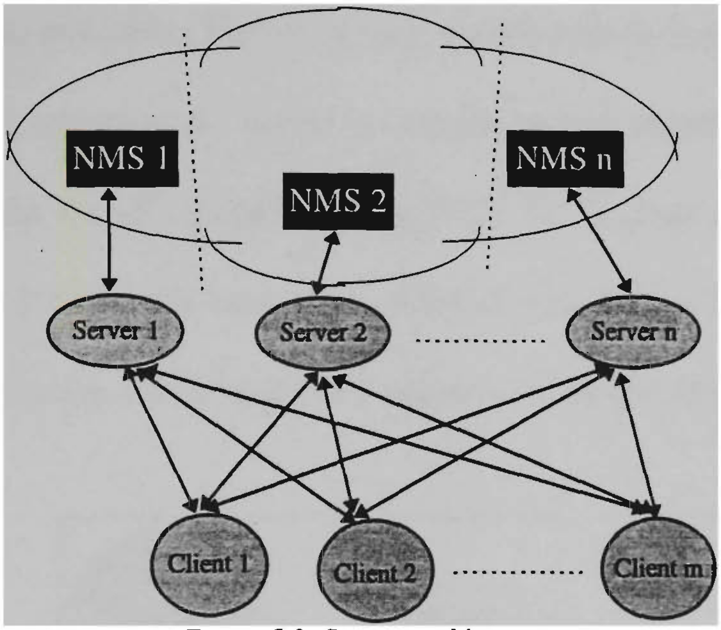

Figure 5.3- System architecture

This architecture has several advantages. Firstly, a universal integrated view of the whole network can be visualised and monitored using standard Web browsers. Moreover, it addresses the problem of multivendor NMS and network elements. The problem is that, NMS tend to provide the best support for network elements manufactured by the same company, but only minimal support for devices from other vendors.

This fact prevents a universal view of the whole network, because each part of it utilises a different NMS. The above architecture allows this integrated view by incorporating the servers as proxies of the network management systems.

However, different network management systems provide different facilities that may not be available to others. Therefore, it is important to determine whether a union or an intersection of these systems should be implemented. It should be noted that it is unlikely and unwise for a management system in one domain to give full access to outsiders. Therefore, only partial access, including monitoring and limited fault management accesses, are granted to outside managers. It is proposed to utilise the full features of the NMS for internal use and provide a common section for external users. This common section has to be standardised, possibly by IETF, to be able to provide a universal view for all users.

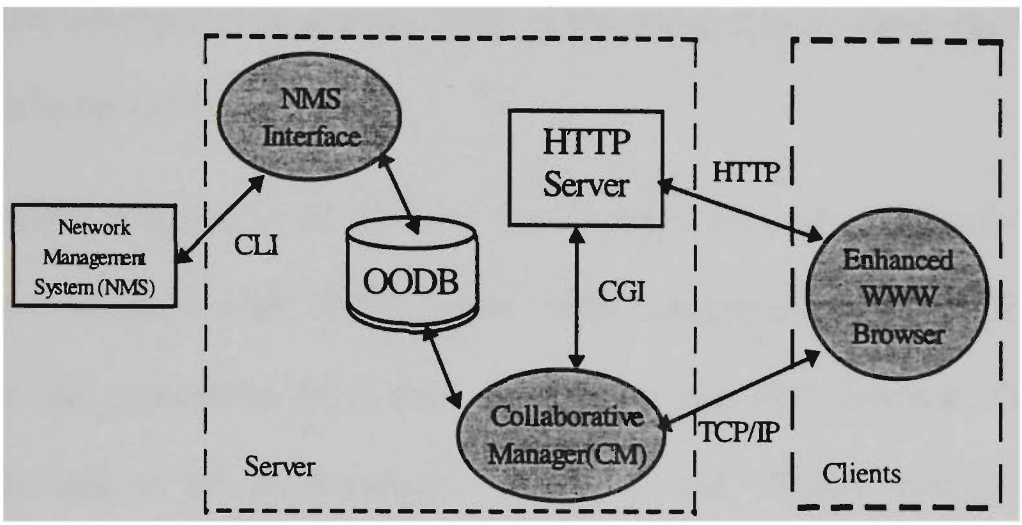

Figure 5.4- Details of client/server communication.

Regardless of the above discussion, each server consists of four parts: NMS interface, Collaboration Manager (CM), an object-oriented database (OODB), and an HTTP server, as shown in Figure 5.4. The NMS interface communicates with the network management system. The communication can be via a shared database or via the command line interface (CLI). The NMS can be any system capable of gathering information from the network elements (NEs), and in the prototype system, is Cabeltron Spectrum. The interface queries the NMS to get updated management information about the status of NEs, and stores them in the

OODB. It also gets update information from the database and sends it to the NMS.

The collaborative manager (CM) is the core of the system. It communicates with the clients directly, or via the HTTP server through a Common Gateway Interface (CGI) script. It also coordinates the collaboration between clients, by collecting the update information from each client, multicasting it to the other clients, and storing it in the OODB.

An example scenario is as follows: the manager uses an enhanced WWW browser to connect to the HTTP server. After authentication, the HTTP server asks for the appropriate view from the CM via the CGI protocol. The CM responds with a set of documents in HTML and VRML formats. These documents contain Java code that firstly, establishes a TCP/IP connection between the client and the server, and secondly, controls the behaviour of the NEs in the client's environment. The user can then navigate into the 3D virtual world, interact with NEs and manipulate the world scene. The position of the navigator and user's manipulations are continually sent to the CM via the established connection. The Java applet also listens to the connection and updates the world scene based on the received data. The GPMI protocol is used for exchanges of information in the established connection.

Whenever the CM receives update data from any client, it broadcast the data to all the other clients and updates the database. If the change has to be incorporated into the NMS, it notifies the NMS interface, as well. In case the user requires more information from the NMS, the request is also recorded in the database. The NMS interface then sends the request to the NMS, and after receiving the required data, puts it in the database. The CM finally sends the result back to the requesting client. The whole scenario is illustrated in Figure 5.4.

All communication between CM and NMS-interface is through the object-oriented database. The rationale behind using this database is that most current commercial network management systems, including Spectrum, do not allow direct access to the management information database. In addition, the database separates the NMS specific part of the server from the CM part, which is mostly NMS-independent. Therefore, adjusting the system for a different NMS requires changing just those parts of NMS interface that sends or receives data to/from the NMS. This separation also makes the implementation easier, modular, and more maintainable.

In addition, as the inter-objects relations and the relation between the objects and other elements of the system are complex, the use of a relational database (RDB) is not suitable. That is, to fulfil a query, the system has to check many tables, which is time consuming. This task is much simpler in an object-oriented database (OODB), as a reference to each object is stored (Permanent ID-PID).

Moreover, it provides facilities for long transactions and large data items. Finally, as an object-oriented language was chosen for the programming, using an OODB combines well with the programming so that in some situations there are no differences between transient and persistent variables.

5.5 Implementation Issues

There are two sets of implementation issues:

1. Issues related to the implementation of different parts of the system (general issues).

2. Issues related to the implementation of collaboration between participants using the GPMI protocol (collaboration issues).

5.5.1 General issues

5.5.1.1 Visualisation

In Chapter 3, the visualisation method was discussed in detail. In this system, a similar method for visualisation is used. That is, the whole network is broken into several virtual worlds, linked together via portals (eg. router and bridges). However, as VRML plugins support texture mapping, this technique has been used to reduce the total number of polygons in each virtual world, consequently, speeding up the rendering.

5.5.1.2 Object oriented database

The object oriented database is based on ODE [Arlein, 1996]. The database can be accessed from C++, and provides libraries and a modified C++ compiler. It also has facilities such as events and triggers that allow effective change verification and notification. The latest version of this product includes a memory-based OODB and the Object Query Language (OQL).

5.5.1.3 Server

The server consists of several parts, communicating with each other via the object-oriented database. The HTTP server is an Apache HTTP server running on a Sun Sparcstation. The CM is a set of C++ programs; most of them act as CGI scripts. A CGI script receives queries from the HTTP server, and returns a valid HTTP output, which is sent to the clients by the HTTP server. In addition to the CGI scripts, a program acts as a server, and is responsible for communication of all collaboration information.

The NMS interface has also been written in C++. It receives the user queries from the OODB, and after consulting with the NMS, puts the requested data in the OODB. In addition, the NMS interface constantly polls the alarms and events from the NMS. Whenever there is a change in the status of the network elements the NMS-interface sends it to the CM to be broadcasted to all interested users.

5.5.1.4 Client

There are many WWW browsers available for most platforms, including Unix, PC and Macintosh. Although good VRML plugins are not yet available for all platforms, most platforms have a working VRML plugin. This is because VRML is a new technology, and it is expected that in the coming year its implementation will reach maturity.

Modern browsers allow the division of the client's window into several frames. This feature has been used to show both text (HTML-based) and graphics (VRML-based) frames within a window. This removes the need for opening extra window and allow simultaneous viewing of both graphic and textual information. The network structure is shown three-dimensionally in the graphics frame, and textual information is displayed in the other frames. Textual information includes some general information about the current view, as well as special data requested by the user, such as lists of alarms and MIB variables.

5.5.1.5 Configuration management

Three-dimensional objects represent NEs in the virtual world. Some objects, such as routers and gateways, act as portals and appear in several views. By clicking on their symbols, another view will open, replacing the current one. This may require connecting to another NMS, which occurs transparently to the user except when authentication is required. For instance, suppose that a domain has a router that connects it to another domain. By clicking on the router's representation, the manager can eventually connect to the other domain, which is being serviced by another NMS. This may require authentication from the other NMS, and may result in refusal, or only a partial view of that domain. This can be seen as an implementation of the X interface in TMN architecture [ITU-T Recommendation M.3010, 1992].

As the client connects to the system from different platforms with different graphics and processing powers, it is desirable that the complexity of graphics is changed to match the capabilities of the display. This means that a powerful machine can have more realistic views with textures and smooth shading, while less powerful computers can view objects as simple primitive geometric shapes. This facility is provided by pre-installing a copy of the symbols that represent the network elements. This feature also significantly reduces the traffic required to transfer the object symbols to the clients.

5.5.1.6 Fault management

Fault information, such as link failure or device breakdown, is shown via colour coding, blinking symbols or using different symbols. For example, a link failure can be represented by a broken line, and a faulty device can blink to attract the manager's attention as shown in Figure 5.6. Moreover, the VRML standard allows directional sounds, which can be used to determine the location of the faults. However, at the time of project implementation, few browsers supported this feature, properly.

5.5.1.7 Performance management

Some types of performance data can be directly represented by the system as part of the virtual world. The thickness of links can represent the amount of load and their colour can show the status of the link. It is possible to show the performance charts using graphical features of Java in the text frame. In the future, it also will be possible to link the system to more sophisticated software, such as spreadsheets, to present the performance data in different manners. Most of these features were left as the future work, because of continuous change of the browsers and the specifications of Java language.

5.5.1.8 Security management

Although, this system is not directly concerned with the security management of the network (because it is the job of NMS not the user interface), there are some security issues that have to be addressed. First, as anybody with a browser can potentially connect to the system, some access control restrictions need to be applied.

Two kinds of security have been implemented for the system: IP address restriction and User IDs. Only browsers from privileged IP addresses can connect to the system, and each user has a unique user ID and password. IP addresses are checked for all transactions, but the user IDs are only checked at connection establishment time. Also, users have different access privileges; some have read/write access, while others only have read privilege. The different security levels are provided by WNMS, (WNMS has full access to the NMS), because it should filter and prepare accesses by mapping user requests to the NMS commands.

It is also possible to use firewalls to prevent system access from outside the domain. Other security measurements, such as content coding for update data, can be considered based on the importance of the management information.

5.5.2 Collaboration issues

In Chapter 4, a generic protocol, GPMI, was developed to address the collaboration in the multiuser interfaces. This protocol has been used to implement the collaboration in the WWW-based Network Management System (WNMS). In this subsection, the details of the implementation are explained.

The login procedure is shown in the diagram of Figure 5.5. It illustrates how GPMI PDUs are used to transfer data between clients and the server. The details of Message PDUs used in this diagram are shown in Table 5.1.

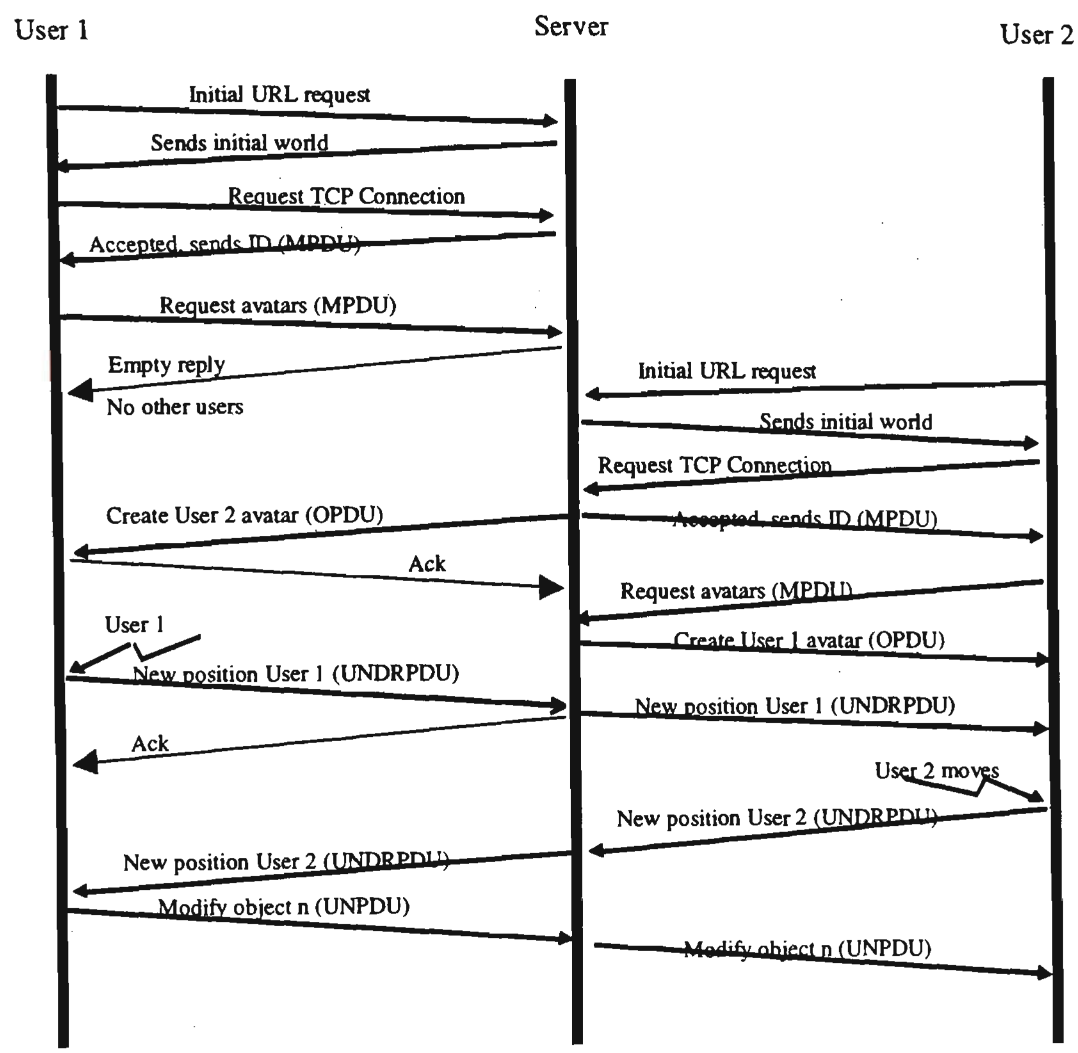

Figure 5.5- Time-space diagram

A user loads the main page, causing the Java applet that is responsible for network connection to be loaded. This applet tries to connect to the CM. If the connection is successful, it sends the server a MPDU notifying that it is ready. [Refer to Table 5.1 for list of *requestType* defined for this system.] The CM responds with the ID of the user in the *message* field. This ID will be used for further communication with the server. The applet then waits until the VRML file is loaded, and the system becomes ready. It then requests the avatars of the other users using MPDU. The CM sends the avatars of those users that are in the user's area of interest, one by one, using OPDU with *action* field set to *CREATE*. The CM also sends the other users the avatar of this user, making them aware of the presence of the new user. The deletion of a user's avatar is done by the CM, automatically, as users change their virtual world, or leave the system. The time-space diagram of Figure 5.5 illustrates this scenario. Note that *requestType* 1 (in Table 5.1) has not been used in the current system, because the initial scene description is loaded via the browser.

requestType	Sending		Receiving	
	meaning	***message*** field	meaning	***message*** field
1	Get World	ID of world to get	Received World	Contents of world
2	Ready	null	Accepted	ID of user
3	Request avatars	null	N/A	N/A
4	Send Alert	null	Received Alert	null
5	Send chat	message	Received chat	message

Table 5.1. requestType definition for Message PDU in WNMS

Initially UNDRPDU was chosen for the transmission of the user's location and orientation. Subsequent tests showed that this was not a good choice. This is because in a user interface, unlike a military simulation system, the speed and direction of the navigator change irregularly and often and the area in which a user is navigating is small. Using a UNDRPDU in this environment causes many sudden jumps of a user's avatar as viewed by other users. Sending many PDUs undermines the main reason for using dead reckoning (saving bandwidth). Therefore, the use of UNDRPDU is not justified, as it adds complexity to the implementation and does not reduce the bandwidth. Therefore, the UNDRPDU was replaced by the UNPDU.

The user's manipulation of other objects was transmitted using UNPDU's, because the data is nearly static. Note that the attributes of objects in the virtual world are related to their real-world counterparts (ie. network elements), and any changes in any part (virtual or real) have to be reflected in the other part. This issue was addressed by introducing the NMS interface as a specific user of the system. That is, the NMS interface sends and receives update notification as do users of the system.

The part of the network hierarchy that each user is managing was selected as the area of interest (AOI). That means that each user only receives update data about the status of network elements in the same LAN segment in which each user is present. However, as some network elements, such as bridges and routers, belong to more than one segment, any change in their status has to be broadcasted to all other containing views.

The communication between users is currently handled through a chat interface. By clicking on the *chat* button (Figure 5.6), another window will open, allowing the user to send messages to other managers, and receive theirs. To attract the attention of a particular user, to start talking, the user should click on the avatar of that user. This causes an alert message to appear for the target user, attracting his attention to the chat window. All of these messages are sent using MPDUs.

The locking mechanism for WNMS is explicit. That is, whenever a user wants to manipulate an object, he has to click on the object to select it. The selected object becomes highlighted. Selecting an object causes a lock message to be sent to the CM using an LRPDU. If the lock is not accepted, the CM sends back a rejected LRPDU, and the object is deselected, automatically. The reasons for selecting explicit locking are the characteristics of network management systems. The sensitivity of information requires a strict consistency between the users and the database. Moreover, a user might need to manipulate a set of objects, and does not need any interference by any other user.

5.6 Observation

Based on the architecture discussed in Section 5.4, and implementation issues explained in Section 5.5, a prototype system was built. The user interface requires a Netscape web browser and the SGI CosmoPlayer VRML plugin. The system was built and tested using a Pentium PC and Windows 95.

A user connects to the system through a specific URL. After authentication, he receives a combination of text and graphics in a framed window, as shown in Figure 5.6. The graphical window shows the structure of the network in a three-dimensional environment. Network elements are represented by 3D objects.

These symbols are chosen so that they can be recognised by the user as representations of their real world counterparts. With this method, the user can understand the type of object at a glance.

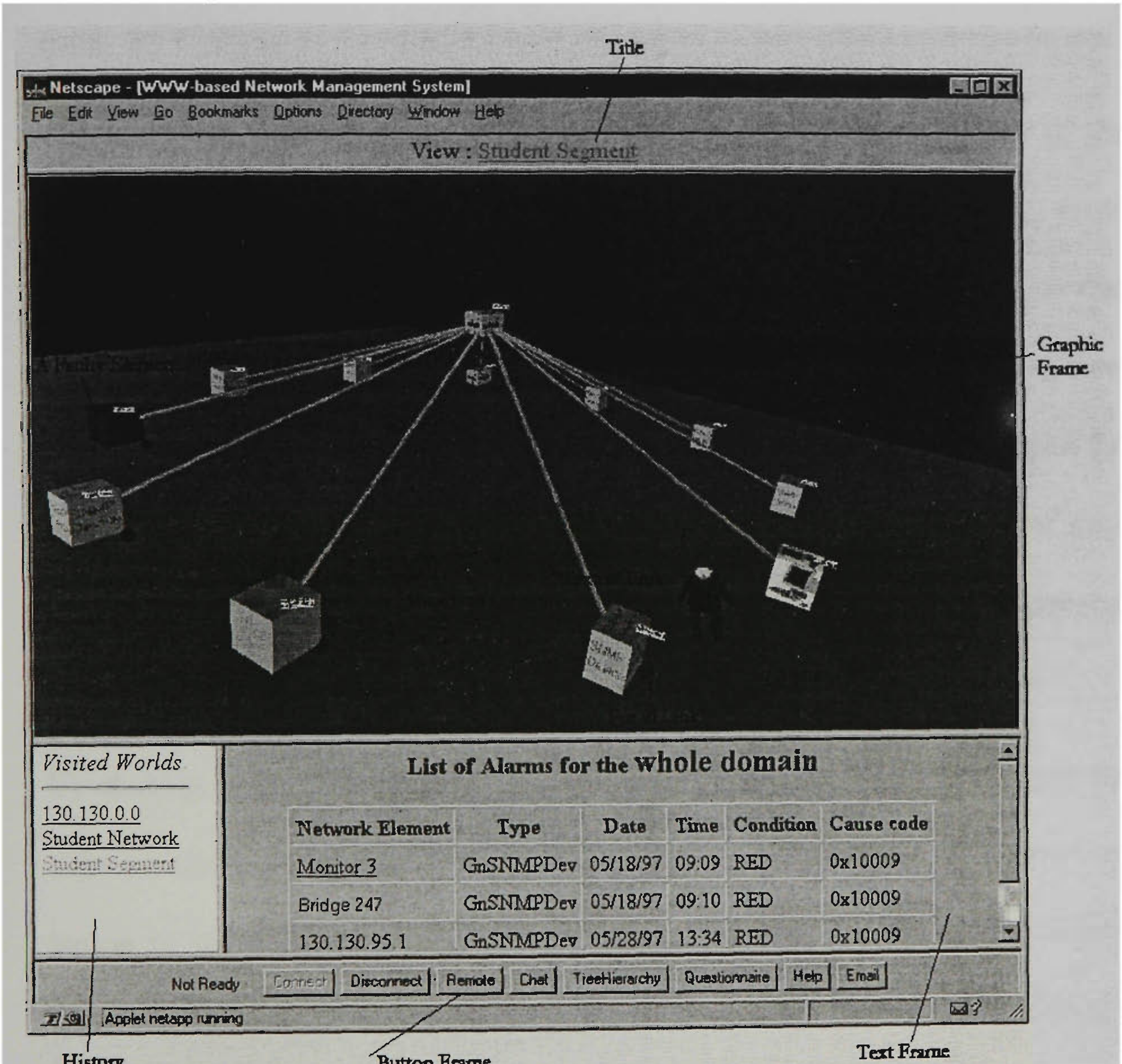

Figure 5.6- A typical view of WNMS.

Each object can have two kinds of links: portal link and text link. Objects that act as links between different views have portal links (shown as small spheres in Figure 5.6). By clicking on the link, another graphical view is presented.

Therefore, the user can navigate the whole network by clicking on portals links within each view.

The textual link is presented by a text line above the object showing the object's name. By clicking on this link, textual information is presented in the *text frame*.

The *text frame* shows some useful information about different sections of the network. After requesting a new view, some general information about that view, including the number of network elements and the number of managers in that view, are shown. Several buttons are also provided, so that the manager can view the alarms and events associated with the current view. For example, Figure 5.6 shows a list of alarms in the *text frame*. Using HTML tags, the name of each object is defined as a link. By clicking on the name, the appropriate 3D view that includes that object appears in the graphical frame.

By clicking on the textual link of an object, some general information about the object is shown in the text frame. The manager can then click on the appropriate button to see the details of the alarms and events associated with the object. Partial or complete Management Information Base (MIB) variables of objects can also be requested.

The user interface has another frame, called *history*, which is shown in the left bottom corner of Figure 5.6. This frame keeps a track of the world visited, so that the user can quickly return to them, by clicking on the name of the desired world.

Only textual communication between managers has been provided. Using this facility, managers can chat with each other, and solve problems collectively. Interviewing local network managers revealed that textual chat is sufficient for most situations. So, because of the complexity involved in the implementation of voice and video conferencing, no more communication was implemented in the prototype system. A chat window is shown in Figure 5.7.

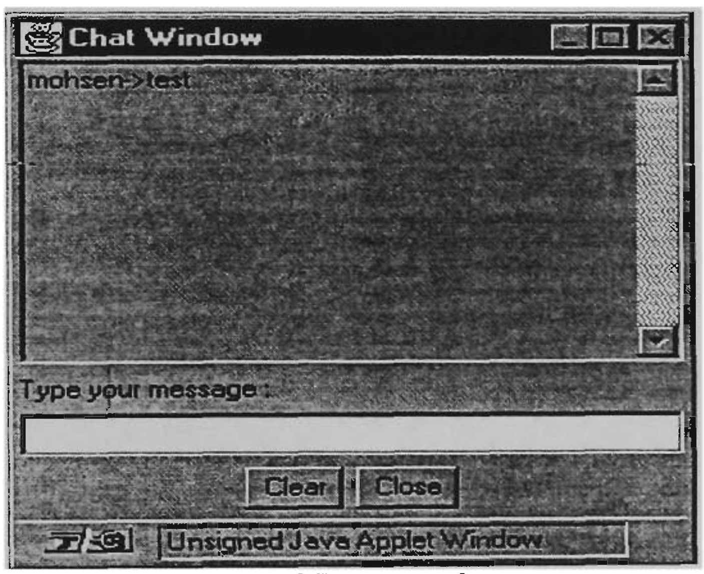

Figure 5.7- Chat window.

The other facilities provided for the system are hierarchical view, remote control and on-line help. The hierarchical view presents the complete hierarchy of the network in one view as shown in Figure 5.8. This view can be used for monitoring of the whole network. Each element is shown as a simple blue box. Faulty devices are colour coded to show the errors in the network. The Cone Tree visualisation method [Robertson, 1991] has been used for this purpose. Therefore, the inter-relation between branches are not shown. By clicking on an element in this view, the appropriate virtual world containing the element is shown in the graphical frame. The hierarchical view appears in another window.

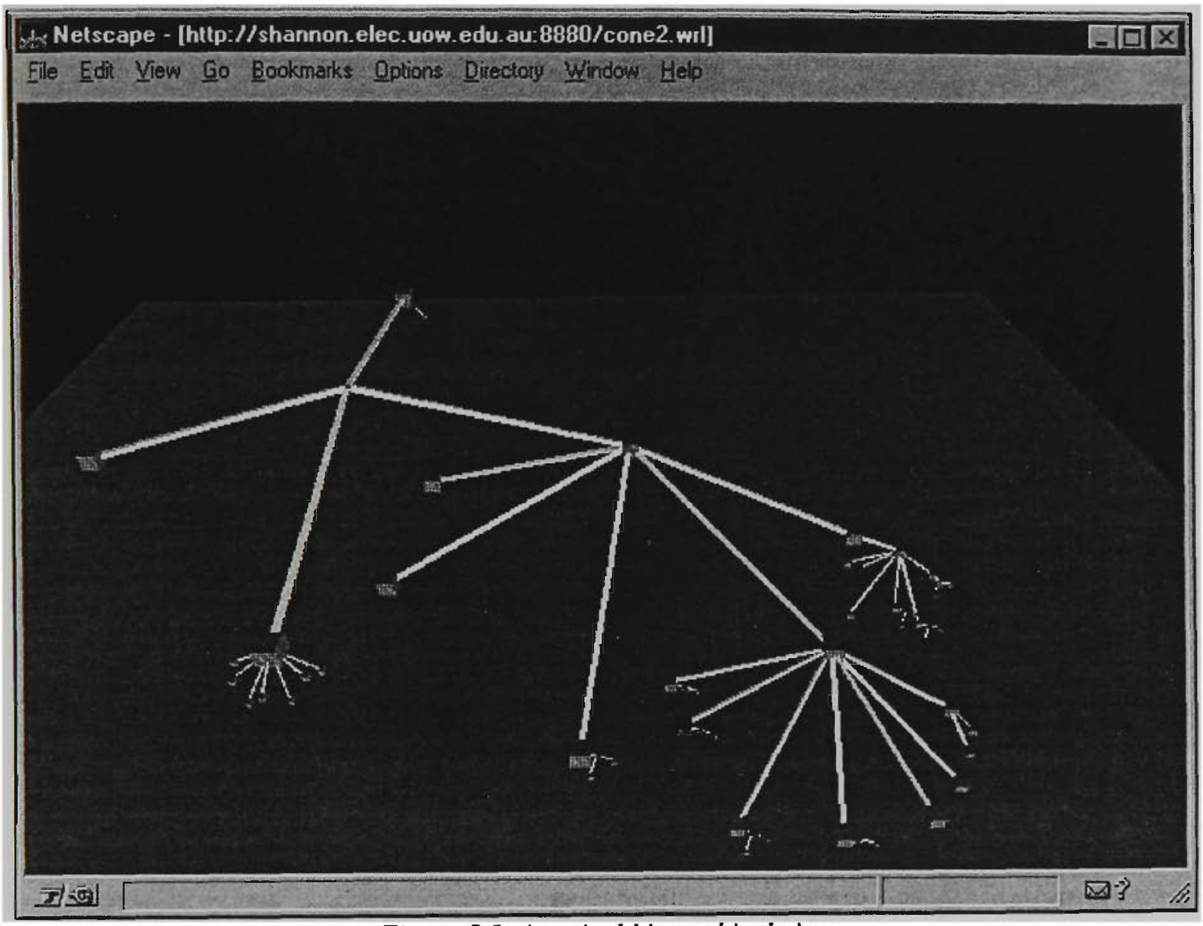

Figure 5.8- A typical hierarchical view.

To facilitate object manipulation in the virtual world, a remote control has been provided. After selecting an object, the remote control allows object repositioning by clicking on the buttons provided for this purpose. The remote control is shown in Figure 5.9.

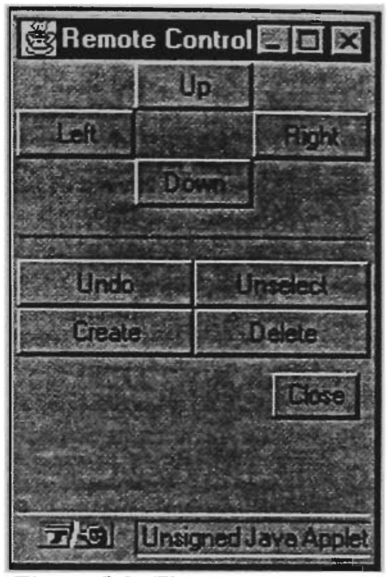

Figure 5.9- The remote control.

An on-line help has also been provided. By clicking on the *Help* button (see Figure 5.6) another window will appear, allowing the user to see the help subjects in a hypertext environment. The help includes an introduction to the system, a pointer to the last updated version of VRML plugin manual, and an illustrated manual for WNMS. A pointer to a list of publications related to the system has also been provided for in-depth description of the system. The first page of the on-line help is shown in Figure 5.10.

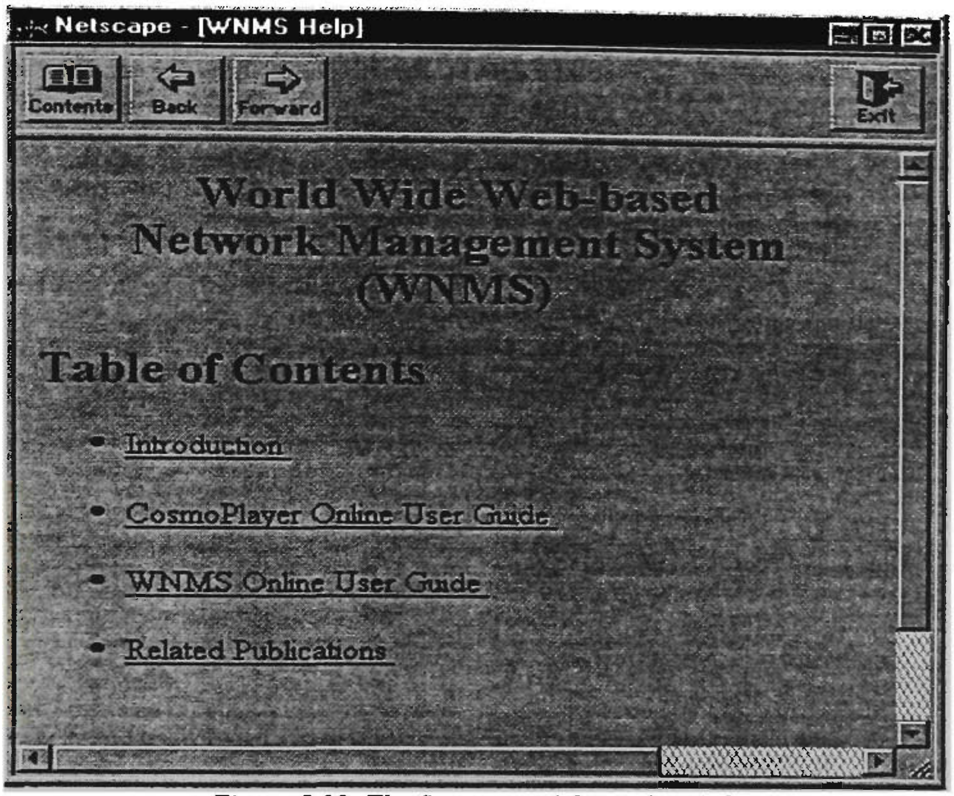

Figure 5.10- The first page of the on-line help

The impression of people seeing and using the system is that, despite its preliminary implementation, its 3D view and mixture of text and graphics gives the manager more flexibility than the available two-dimensional traditional network management systems. Incorporating voice communication between participants and using more realistic and complex scenes could further improve the system.

5.7 Qualitative Evaluation

In this section, a qualitative evaluation of the system is presented, showing its strengths and weaknesses. This system is also compared to the immersive system discussed in Chapter 3. A more rigorous evaluation and comparison of this system with traditional user interfaces is provided in the next chapter.

One of the main features of the Web-based Network Management System (WNMS) is its platform-independence. The managers can connect to the network, from any computer at any point, either remotely or locally, from a notebook or an expensive graphical workstation, and utilise the full capabilities of the system. The managers can take mobile computers with themselves to the fault locations, and collaborate with the managers at the central station to fix the fault quickly, and with greater confidence. This is in contrast with using a traditional VT-100 terminal at the fault location, and calling the central station for consultation or sending test signals.

In both developed systems, the three-dimensional view of the network hierarchy and additional navigational facilities increase the visualisability of the network management information. The greater visualisability means the lower probability of error and miscalculation by the manager. This directly increases the network survivability and reduces down time.

Network repair technicians acknowledge that there are some situations where one needs to be in at least two places at once. The collaboration feature of WNMS helps to address this problem.

In comparison with the immersive system, clearly that system provides a much higher level of immersion. Immersion could provide a feeling of directness, so that in times of stress, users only need to think of 'what to do' without needing to consider 'how to do it'. This means that quicker and better actions can be chosen in times of stress.

Instead, WNMS benefits from some of the advantages of WIMP interfaces. Multiple frames consisting of 3D graphics and text carry more information than a pure graphical one. Also, as mentioned earlier, in a network management environment, there is plenty of useful textual information that cannot be translated efficiently into graphical symbols. WNMS can easily show the information in the text frame, while the display of text in a graphical environment is more difficult.

Another point of comparison is interaction of the users with the system. The level of interaction in an immersive system is quite high. Utilising VR gloves and other 3D input devices such as Cyberman and a 3D mouse, makes navigation in the 3D world quite easy. On the other hand, most WWW browsers only use a mouse for interaction. This causes some confusion as the user has to switch between different modes of movement, eg. *walk* versus *examine*. However, it appears that future browsers will support more input devices.

Platform dependency and tethered operation are other weak points of immersive system. Immersive systems are mostly platform dependent, and developing a system for a special platform requires a special set of libraries and utilities that might not be available for other platforms. In addition, most VR devices are platform dependent. The existence of these devices reduces the mobility of the immersive system, as well.

The other advantage of the WWW-based approach is higher accessibility to the network management system. The prices of browsers are so low that most computers have a copy of them installed. Therefore, managers can use virtually any computer in the network to connect to the system, and benefit from the graphical user interface, to manage the network, remotely. With collaboration added to the system, the scope and level of management go far beyond current systems.

Both systems have short learning times, though for different reasons. The learning time in immersive systems is short because the interface is designed to behave naturally. The short learning time in WNMS, on the other hand, comes from the fact that most users are already familiar with the browser based user interface. This is in contrast with most commercial systems where operator training and remembering the user interface is times of stress is a major problem. A short learning time for the user interface allows training to focus only on the issues related to network management.

Furthermore, the trend in network management is moving towards using HTTP and Web technology, combined with SNMP or independently, for device polling and status notification [Wellens, 1996]. Therefore, WNMS can be seemlessly expanded to directly contact the network elements. In fact, the trend towards Web-based network management is so high that some experts believe that "the network management platform of the future may only have Web-based user interface" [Bruins, 1996].

5.8 Conclusion

The design and implementation of a WWW-based three-dimensional interface, in which operators can effectively and collaboratively manage distributed heterogeneous broadband networks, was discussed in this chapter. The system uses a client/server approach to implement a TMN-style network management system. The server part uses an object-oriented database to exchange information between participants and the network management system. The client is a WWW browser enhanced with VRML and Java capabilities.

The prototype system provides some useful features. Unlike most commercial management systems that need expensive graphical workstations for network visualisation, in WNMS, managers can connect to the system, and do full network operations from any location in the network using virtually any computer. However, more powerful computers can deliver a very realistic view featuring texture mapping and smooth shading, and less powerful machines use rather primitive 3D icons, with a reasonable speed, to visualise the network.

The three-dimensional and collaborative environment created by this system, firstly, provides a potentially greater visualisation of the system, and secondly allows real-time communication between managers, which is necessary for management of complicated and flexible broadband networks.

Shorter learning time, provision of a universal integrated view for heterogeneous networks and lower prices are other advantages of the WWW-based user interface. In addition, the fact that network management is moving towards web-based approaches allows this system to be more efficiently integrated with the network management system. In fact, some NMS providers, such as Cabeltron, Bay networks and Computer Associate, have already added Web-based interfaces to their products.

Because of the object-oriented approach used in the design and implementation of this system, the system can easily be adapted for other systems. That is, by replacing the NMS interface, by the appropriate interface, and minimal changes to other parts, it can be used for similar multiuser environments.

Chapter 6

Quantitative Evaluation and Comparison

6.1 Introduction

In this chapter, by using more rigorous methods, a quantitative usability evaluation and comparison is presented. As part of the comparison, WWW-based network management system (WNMS) is compared with the traditional WIMP based user interfaces for the Cabeltron Spectrum network management system.

Initially, the evaluation and comparison methodology is discussed. Based on the discussion, several evaluation methods are selected. The structure of the user satisfaction survey is then presented. This is followed by the evaluation of the systems for both specific network management issues and general user interface issues. Finally, the results are discussed and a conclusion presented.

It should be noted that the WWW-based system is a prototype while Spectrum is a well-developed system and represents one of the better user interfaces for network management in the market. That is, WNMS is one possible way of designing Web-based 3D systems and its disadvantages are not necessarily inherited from its three-dimensional display. Therefore, its merits show the merit of the architecture, and its pitfalls provide background for further investigations.

6.2 Experimental Methodology

The usability of an interface is a key concept in the study of human computer interaction (HCI) as it reveals the problematic areas and thus makes the system safe, easy to learn and easy to use [Preece, 1995]. So, it is desirable to evaluate and measure the usability of a system. This evaluation can be performed using several of the qualitative or quantitative methods discussed in Chapter 2. The advantage of quantitative evaluation is that it provides numeric scores that can be used to compare two systems. An analogy from circuit theory would be comparing filters based on their Q factor. The drawback of simple quantitative analysis for complex systems is that it is usually difficult to express some attributes by values. Therefore, each of these evaluations has its own domain of usage.

In most cases, the usability of a system is measured, using *de facto* standard techniques (for example, usability engineering and SUMI questionnaire), and is compared with the values already obtained for similar systems to reveal the strength and weakness of the developed system. However, little research has been done regarding the usability of network management user interface systems. Therefore, no hard figures for evaluation of these systems are available for comparison. To remedy this lack of information, Spectrum was selected as a representative of the traditional network management systems, and the evaluation process was applied to both WNMS and Spectrum. This adds to the complexity of the already difficult task of evaluation, as some other factors, such as fairness and provision of a proper environment for each system, have to be taken into account.

While there are standard guidelines for the design of general user interfaces, (eg. ISO 9241), there is no set of guidelines in the literature regarding the specific requirements of network management systems. Therefore, the design and comaparison guidelines also have to be developed.

Regardless of this lack of information, the experiments pursue two main goals, in general:

1. comparing the design architectures of both systems; and
2. determining the usability of WNMS.

The former goal is aimed at ascertaining if collaborative 3D user interfaces are superior to traditional user interfaces for network management. The latter goal provides further explanation for the result of the first goal, and reveals usability problems requiring further investigations.

As shown in Table 2.2 of Chapter 2, observation and monitoring, user opinions and predictive evaluation methods can be used to compare two systems. Here, a set of these methods were used supplementarily to perform the comparison. Direct observation, structured and unstructured interviewing and questionnaires were used to elicit more information from the subjects. In addition, heuristic and cognitive walkthrough methods were used to develop a guideline for the comparison. The workload was also measured using the NASA-TLX survey [Biferno, 1985].

The remainder of this chapter is structured as follows: The rest of this section discusses the way the subjects were chosen and the statistical methods used to analyse the results. The structure of the user satisfaction questionnaire is explained, in the next section. Using the result obtained from the survey, interviews and observations, the usability of WNMS is discussed.

The section on the comparison is divided into specific network management issues, and general user interfacing issues. For instance, 'guiding fault management' is categorised as a network management specific issue, while workload measurement has been considered as a general user interface issue. Each set of issues is discussed in a separate subsection.

6.2.1 Subject Selection

Several evaluation methods are required to assess different attributes of the systems. In addition, each attribute should be evaluated by a set of suitably qualified subjects. For instance, while learnability testing should be conducted upon novice users, experts should conduct the heuristic evaluation. Overally 15 subjects were selected for the experiment and were categorised into three subsets, as indicated below:

1. Novice users: The novice user set consists of five postgraduate students. While they were familiar with networking and had been using WWW for a long period, they did not have any network management background. In addition, they had no or little

experience with VRML. The reason no undergraduate students were chosen was that VR and 3D user interfaces are attractive, especially for young people. This attraction could have influenced them to make a biased decision. In addition, more time would have been required to teach them how to use the systems, particularly Spectrum.

2. Intermediate users: This group consists of five postgraduate students. Although they had little network management experience, they were familiar with network management issues. Most of this group had attended network management courses, and were modestly familiar with Spectrum. Some of them also had experience with VRML.

3. Expert users: The expert set consisted of five people experienced in campus-level network management. They were familiar with Spectrum, and had some experience with VRML.

The subjects performed the evaluation individually. Each subject used each system for 30-40 minutes. The systems were evaluated in a random order. Before using any system, the subjects were given a 10-minute tutorial, and they were provided with assistance during the evaluation. Upon finishing each part, the users were asked to complete a NASA-taskload index questionnaire and interviewed. They also filled a user satisfaction questionnaire at the end of the experiment. It should be mentioned that while the sample set is small, it is difficult to find qualified subjects for a broader trial.

6.2.2 Statistical Methods

Several statistical methods were applied to analyse the results obtained from the questionnaires. For the user satisfaction questionnaire, the average mark of each section was normalised (0-100) to show the amount of satisfaction of the subjects for that section. A 95 percent confidence interval (α=0.05) was also calculated for each value. A value higher than 50 for any section means that users liked that feature of WNMS.

For the comparison section of the user satisfaction questionnaire, however, a two-tailed *Z-test* was calculated. This test generates a standard score for *x* with respect to the data set, and returns the two-tailed probability for the normal distribution. The *Z-test* can be used to assess the likelihood that a particular observation is drawn from a particular population. Therefore, this test can determine if the subjects preferred an approach. The *Z-test* is calculated using the following formula:

$$Z-Test(array, X) = 1 - NormalDistribution\left(\frac{\mu - X}{\sigma \div \sqrt{N}}\right)$$

where:

array = array of data against which to test *X*.

X = value to test and equals to 50 for the test.

σ = standard deviation of the samples

μ = mean of the samples

N = number of samples

For the TLX survey, a one-tailed paired Student-test (*t-test*) is calculated. This test, also called the dependent-directional *t-test*, is used when the same subjects are exposed to two experiments (or before and after an experiment), with the aim of proving that one experiment has better results than the other [Ary, 1996]. That is, it is desirable to prove not only that the experiments have different results (reject the null hypothesis), but also a particular system is better than the other.

In this thesis, the *z-test* and *t-test* results are shown in terms of the confidence level that the *null hypothesis* is rejected. For instance, if an statement is followed by $p<0.05$, it means that the probability that the statement is false is less than 5%.

6.3 User Satisfaction Questionnaire

The questionnaire has been designed to evaluate WNMS and provide a comparison between WNMS and a traditional interface. It contains 47 bipolar descriptor 5-point rating scale questions (Figure 6.1) categorised into eight sections. These sections are titled as general, navigation, interaction, collaboration, textual data, learnability, overall performance/satisfaction and comparison. The questionnaire was developed and administered as an HTML document (Figure 6.1).

The general questions do not provide any information regarding the evaluation of the system. They are used to validate the answers and to categorise the subjects. These questions ask the user about the user's background and experiences in both network management and virtual reality systems.

Under navigation related questions, users are asked to express their views on the aspects of the system that helped them understand what was happening in the network. The subjects should express their views on whether the network and scene help them properly visualise and monitor the network. The subject should also assess if movement between different views could be easily accomplished.

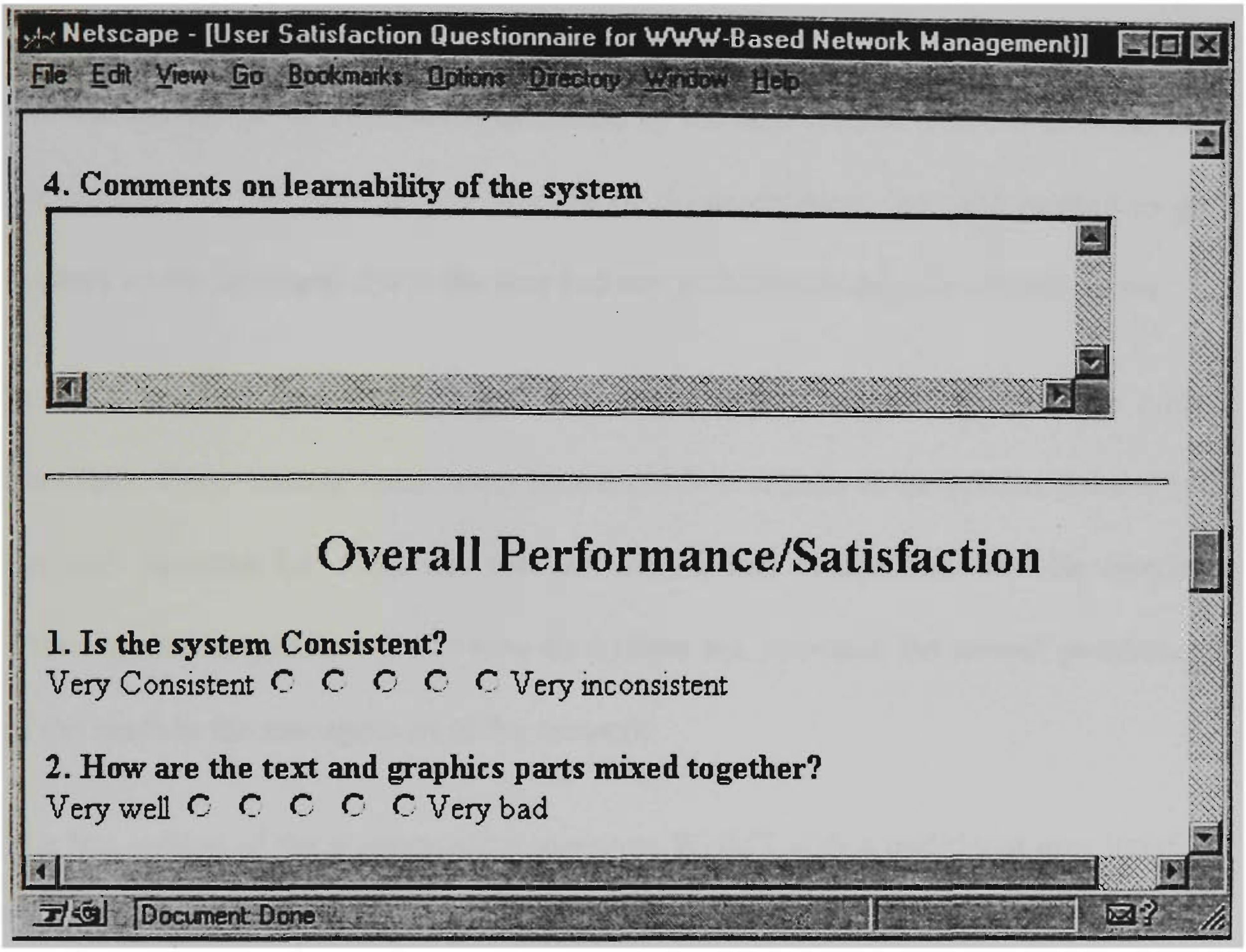

Figure 6.1- A view of part of the user satisfaction questionnaire

In the next set of questions, the subjects assess whether the user interface allows them to easily manipulate objects. The questionnaire asks them to indicate whether the user interface responds predictably and consistently to the inputs.

The collaboration features of the system are assessed by the next series of questions. The users also evaluate the provision of chat among the users. Moreover, they express their view about the usefulness of other features, such as telephone conversations and other forms of teleconferencing.

The next set of the questions asks the user if the interface provided enough textual information. In addition, the user is asked to assess how effectively the textual data is presented.

The learnability of the interface is evaluated by the next section. Three important issues are assessed in this section: the amount of documentation, the time needed to gain mastery of the interface, and if the user had any problems finding the correct action.

In the overall performance/satisfaction section, subjects respond to the most critical questions. These include consistency between different parts of the system, suitability of the user interface for every day use and overall user satisfaction with the interface. These questions clearly indicate how the system has increased the overall performance of the users in the management of the network.

The last section of the questionnaire compares WNMS with a traditional user interface. The questions ask the user to separately compare the systems in terms of visualisation of the hierarchy and management information, ease of navigation and interaction, learnability and overall performance.

Besides the questions, at the end of each section, the users are asked to provide open-ended comments. The complete text of the questionnaire can be found in Appendix D.

6.4 Evaluating WNMS

After analysing the results of the questionnaire, a score is obtained for each section, as explained in Section 6.2.3. All scores are normalised between 0-100, and are tabulated in Table 6.1. These scores demonstrate the usability of WNMS. It should be noted that no score for the collaboration section has been calculated due its nature (see Section 6.5.2.5).

	Navigation	Interaction	Textual Data	Learnability	Performance Satisfaction	Comparison vs. Spectrum
Score	74	79	73	82	82	83
Confidence Interval	4	4	5	4	4	3

Table 6.1- The result of questionnaire evaluating WNMS.

As all scores are well above 50, it is a clear that the subjects liked the 3D Web-based user interface. In particular, a high mark for performance/satisfaction (82 ± 4) shows the users were quite satisfied with the system. This fact also was verified in the post-experiment interviews. The average score of (83 ± 3) for the comparison section clearly demonstrates the users' attitude toward both systems.

However, after performing the experiment and analysing the result, it was argued that as the subjects did not evaluate Spectrum separately, and only compared it with WNMS, the result might be biased. To answer this concern, the evaluation part of the questionnaire was slightly modified, and the expert subjects were asked to evaluate Spectrum. This trial was performed a few month after the initial survey. The result of this evaluation, as shown in Table 6.2, confirmed that result of initial comparison.

	Navigation	Interaction	Learnability	Performance Satisfaction
Score	39	36	25	42
Confidence Interval	8	10	7	10

Table 6.2- The result of questionnaire evaluating Spectrum.

To avoid repetition, a more detailed discussion of the results is left for the comparison and discussion sections. That is, the results of the statistical analysis of the questionnaire and interviews are used to further explain the results of the comparison measurements.

6.5 Comparison

In this section, several measurement methods, in addition to user opinions, are presented to provide a fair comparison between WNMS and Spectrum. The evaluation is based on the specific network management issues and general user interface issues. In the absence of any standard or guideline, the network management issues first have to be determined. Therefore, guidelines are developed and then both systems are compared with each other using the guidelines.

6.5.1 Guideline Development

As mentioned earlier, there is little published literature on the specific requirements of network management user interfaces. This is in part due to the multi-disciplinary nature of Human Computer Interface (HCI), thus requiring a multi-disciplinary team to address the relevant issues [Mandich, 1992]. Terplan has categorised the results of a survey of some network managers to determine the general requirements of network management systems [Terplan, 1992]. The survey contains a relatively long list of general requirements, and includes some user interface issues. However, as the focus of the survey was not on the user interface, it seemed quite likely that some important factors had been missed. Moreover, the survey is a few years old, and the evolution of the technology might have forced further requirements.

To probe further, it was decided to use the guidelines mentioned in the above survey as a base, but to add or delete items as required. To achieve this goal, the expert group of experimental subjects was interviewed. The interview used a combination of structured and unstructured questions to elicit more information. This resulted in the addition of two factors; low bandwidth requirement and high accessibility. The non-user interface guidelines were deleted, and others were slightly refined from the items mentioned in Terplan's survey. The additional items, represent a greater need for accessing the network management information from geographically dispersed locations. For example, managers want the capability of connecting to the network from home and accessing the management information in a graphical format, rather than via VT100 terminals. The complete set of the developed guidelines is:

1. Guiding fault management
2. Integrated universal interface
3. Straight forward interface (easy to learn)
4. Centralised view with distributed management scheme
5. Co-operation among users and between users and service providers
6. Low bandwidth requirements
7. High accessibility.

6.5.2 Network Management Issues

6.5.2.1 Guiding fault management

Fault management is one of the most important tasks of a management team. However, because of security issues, most Internet connected network elements do not allow remote configuration. Therefore, current network management systems only provide fault reporting rather than actual fault management, and current network management systems guide the operator to detect and fix the fault.

It is quite difficult to directly measure which system provides better guidance for the network manager. It requires utilisation of interpretive techniques. Referring to Table 2.2 of Chapter 2, interpretive methods are not appropriate for comparing systems, as those techniques provide qualitative results not suitable for comparison.

Providing a synthetic environment would be expensive. It is also difficult to assure subjects in the synthetic environment have gained the expertise that they experience in a real environment. In addition, it is difficult to generalise the result obtained for a small and experimental network to a large-scale network. Considering all these limitations, it was decided to use indirect measurement. That is, instead of generating some intentional faults and observing the amount of guidance that each system provides, by measuring the Mean Time To Diagnose (MTTD) and Mean Time to Repair (MTTR) for example, it is better to find out the important factors for fault guidance, and measure them.

In this regard, Crutcher has discussed that problems experienced with traditional network management user interfaces stem from their low level of visualisation and interaction in these environments [Crutcher, 1993]. This fact has been verified by other researchers, too [Kahani, 1995; Chen, 1996; Deri, 1997]. Therefore, visualisation, navigation and interaction factors were considered suitable for comparison.

The statistical analysis of the results of the survey shows that subjects strongly favoured WNMS over Spectrum, in terms of visualisation ($p<0.01$) and navigation facilities ($p<0.005$). However, the difference in terms of interaction facilities, though slightly better, is not statistically significant ($0.05<p<0.1$). This result was partly expected, as a mouse does not provide good facilities for interaction in a 3D environment. In addition, the beta version of the VRML plugin used for the prototype did not allow utilisation of all VRML capabilities for interaction.

Considering the result, WNMS is expected to provide better facilities for guiding fault management. However, as the exact amount of contribution of the above factors is not quantitatively available, it is not possible to state how much improvement has been achieved.

6.5.2.2 Integrated universal interface

Today many commercial network management systems are available. Most network management system providers are also suppliers of network elements. The competition among companies has caused each NMS to provide the best support for the elements from the same company (often using proprietary extensions to standard protocols or opaque data types) and little for others. Although, most systems provide facilities for a user to define models for other elements, this task is usually not simple or cheap.

This fact has forced users to purchase the network management system that provides the best support for their existing network elements, producing a heterogeneous set of network management systems even within a campus. The long training period for each system and lack of compatibility among the systems has produced isolated islands of management information, not accessible by others. This problem is known and has been acknowledged by most managers interviewed by the author throughout the investigation. More detailed discussion of the problem caused by this incompatibility can be found in [Hong, 1997].

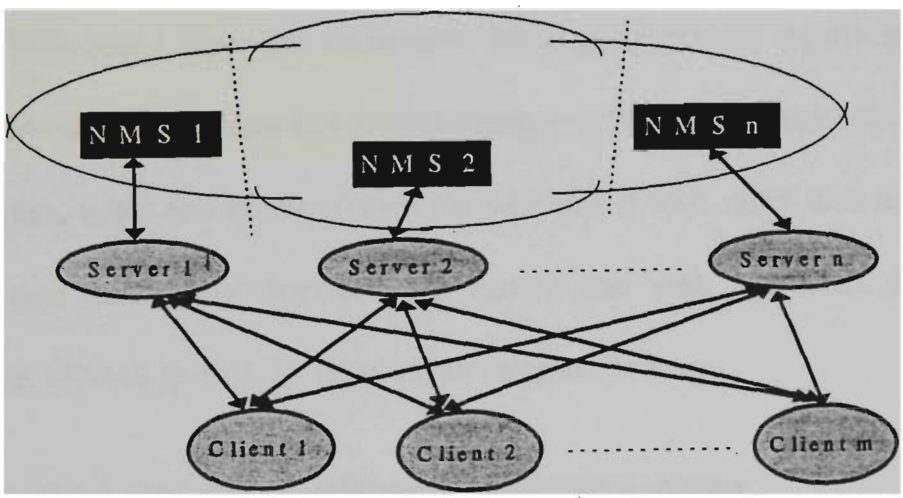

Figure 6.2- The architecture of WNMS.

WNMS addresses this problem by providing a WWW-based universal interface. As shown in Figure 6.2 (Figure 5.2 reproduced), the WNMS servers act as proxies to network management systems. That is, the servers get the management information from different and heterogenous NMSs, and provide a universal user interface. With this feature, any manager can access and visualise any part of the network without having to install a new interface and learn its operation.

In addition, WNMS allows utilisation of other Web applications, such as email, Java applets, etc. For instance, Java applets can be used to show performance information in real-time. This also provides an integrated familiar environment for users, reducing their training period.

6.5.2.3 Learnability

Learnability is another important factor in network management systems. Although, nowadays most users are familiar with WIMP user interfaces, training staff for efficient use of management software still requires a significant amount of investment [Jander, 1996]. A WWW-based approach minimises the training period, significantly. In the experiment, all subjects felt they had gained mastery of the WNMS user interface in less than 30 minutes, while the second survey showed that it took more than a day for the subjects to gain mastery of Spectrum. In comparison with Spectrum, the subjects strongly chose WNMS ($p<0.005$), as an easier interface to learn.

6.5.2.4 Centralised view with distributed management scheme

Although most commercial network management systems provide some facilities for distributed management, their application is limited due to the heterogeneity problem discussed in Section 6.5.2.2.

WNMS provides both interim and permanent solutions for this problem. The interim solution allows current network management systems to be used by adding a Web-based front-end, as explained in Chapter 5. This option employs the current infrastructure and trained managers. It also provides a centralised view of the network state, as all systems support a universal interface.

The long-term solution includes incorporation of the server part of WNMS with commercial network management systems, or using pure Web-based systems. Indeed, some systems, such as Spectrum, HP OpenView and Unicenter TNG, have incorporated some WWW interfaces, though limited in functionality and solely HTML-based.

Pure Web-based solutions are also limitedly available. The introduction of the Java Management API (JMAPI) [Sun JMAPI, 1996], and the HTTP-based network management standard [Deri, 1996] have paved the way for full-scale network management systems based on WWW technologies. In addition, the architecture proposed by Web-Based Enterprise Management (WBEM) [WBEM Consortium, 1996] provides an industry standard framework for managing heterogeneous networks, systems and applications using Web browsers. The user interface features of WNMS (especially its three-dimensionality) can be incorporated into this framework to provide a complete network management system.

6.5.2.5 Co-operation

The widespread deployment of virtual private networks based on ATM technology has resulted in a high-level of interdependence among logically separate networks. Consequently, co-operation and collaboration between managers of the logically separate networks and the physical network is often required to address fault, performance problems and other management issues. The ATM Forum, recognising this need, has released the Customer Network Management (M3 Specification) document [ATM-CNM, 1996]. The CNM specification provides information about retrieving performance monitoring data (class I) and virtual connection configuration information (class II), cooperatively. However, it is not clear how this co-operation should be implemented and used.

The experimental subjects, especially the expert group, acknowledged the usefulness of collaboration among managers within a domain. However, they were not quite sure about the possible methods of co-operation among management groups of different sites. Consequently, most subjects refrained from rating the suitability of the facilities provided by WNMS for collaboration purposes.

The subjects also believed that text-based chat is enough for inter-user communication ($p<0.05$). In addition, the subjects did not think that utilisation of other teleconferencing facilities would result in a better collaboration ($p>0.05$). In interview, the network managers mentioned that those facilities distract them from doing their job.

6.5.2.6 Bandwidth requirements

Bandwidth requirements for user interfaces is becoming an important factor in network management systems. Managers want the capability of accessing management information from geographically dispersed locations or through low-speed connections. Currently this task is achieved using text-based dumb terminals and the command line interfaces of management systems. This provides such a poor and difficult interface that all of the managers interviewed mentioned that they would rather use simple *ping* commands. However, this facility is impaired. Should the size of the network increase and the configuration change too often, the managers cannot remember the network configuration.

The problem stems from the fact that most network management systems use a method similar to X-Window to update the client's screen. In this method, all modifications to a screen have to be transmitted by the server. As shown in Figure 6.3, for Spectrum, scrolling the window by the user causes an update message from the server to repaint the window.

	Startup	**Navigation**	**New view**
WNMS	82.5 Kbytes	Collaboration data: 381 bytes/sec.	940 Kbytes
Spectrum	474 Kbytes	Idle (mouse movement): 366 bytes/sec.	14.8 Kbytes
Ratio (WNMS/Spectrum)	17 %	Not Applicable	1.6 %

Table 6.3- Comparing bandwidth requirements.

To show the difference between WNMS and Spectrum, the bandwidth usage for the user interface part of each system were measured and are shown in detail in Figure 6.3 and Figure 6.4, and comparatively in Figure 6.5. The figures show the bandwidth requirements of each system, during startup, navigating within a view and navigating into another view. The same views with the same number of network elements have been used for both systems. The traffic produced by both systems is deterministic. Background network traffic during the tests was minimal and has been filtered out.

Table 6.3 shows that although the initial bandwidth requirement of WNMS is 17% of that of Spectrum, the subsequent views require only a negligible amount of bandwidth (1.6%), compared to Spectrum. Note that the second view has three times more network elements than the initial view. Therefore, a considerably higher bandwidth is required for Spectrum. However, for WNMS, as downloaded files stay in the browser's memory and disk cashes (due largely to the heavy optimisation of WWW browsers for low bandwidth environments), they can be reused for subsequent views. Additionally, it is also possible to pre-install most parts of the Java applet used by WNMS, and all of the element shape files on the local hard-disk, reducing the initial download bandwidth, dramatically (up to 60% for the test case described above).

The bandwidth requirement for navigation within a view is also an important factor. For Spectrum, this is the data required to repaint the part of the display that has changed (due to mouse movement or when part of a window is exposed by closing another window) or scrolling to view other parts of the view. For WNMS, it includes the collaboration information exchanged among users. As shown in Table 6.3, the typical bandwidth required for mouse movement in Spectrum is nearly the same as the collaboration information for a single user environment in WNMS. Other actions, such as scrolling require more traffic (eg. 14 bytes/sec for each click on the scroll bar).

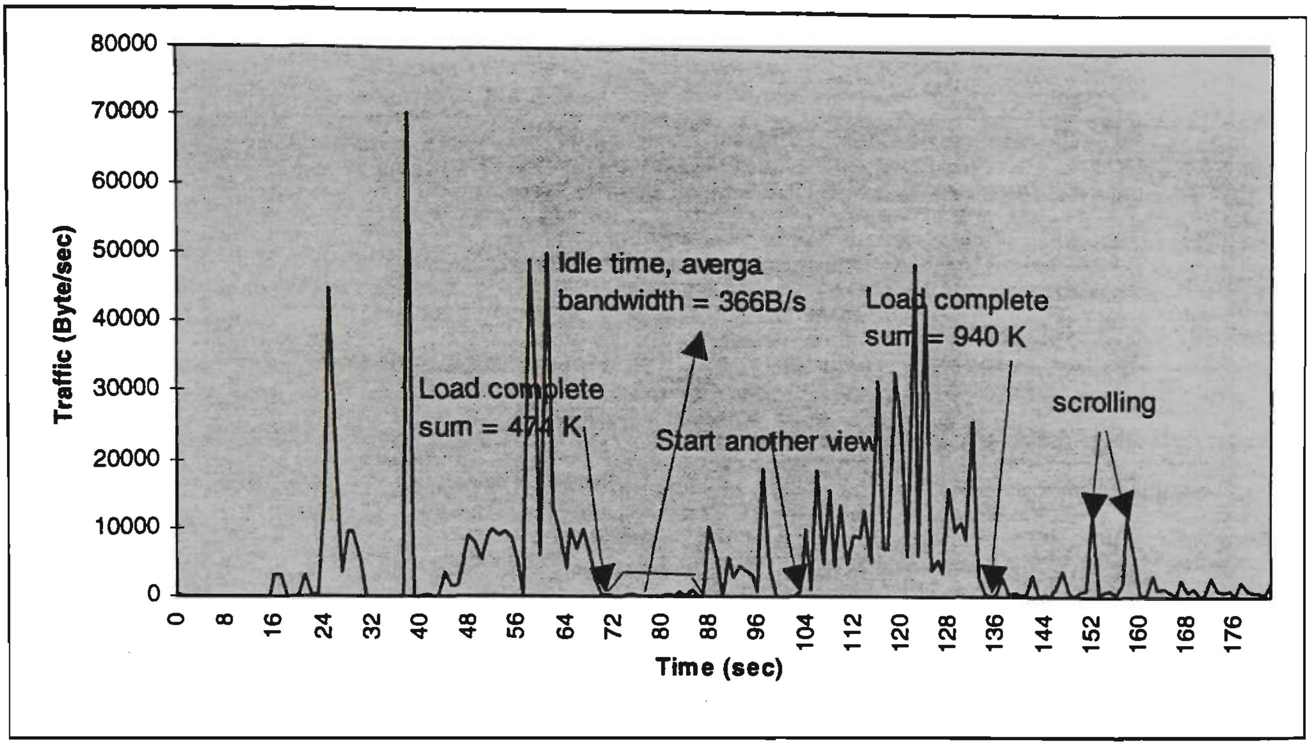

Figure 6.3- Bandwidth requirement of the Spectrum user interface.

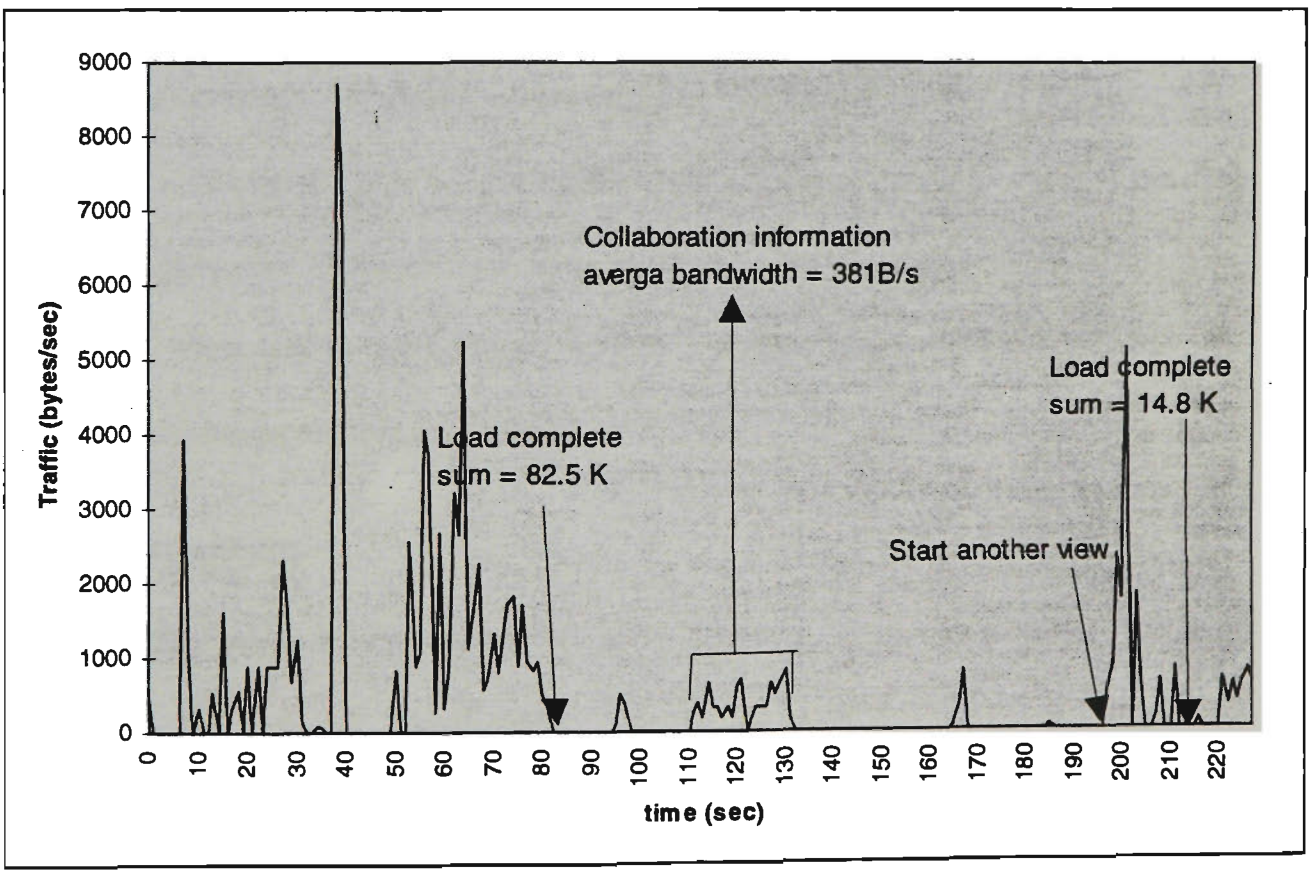

Figure 6.4- Bandwidth requirement of WNMS user interface (1 user)

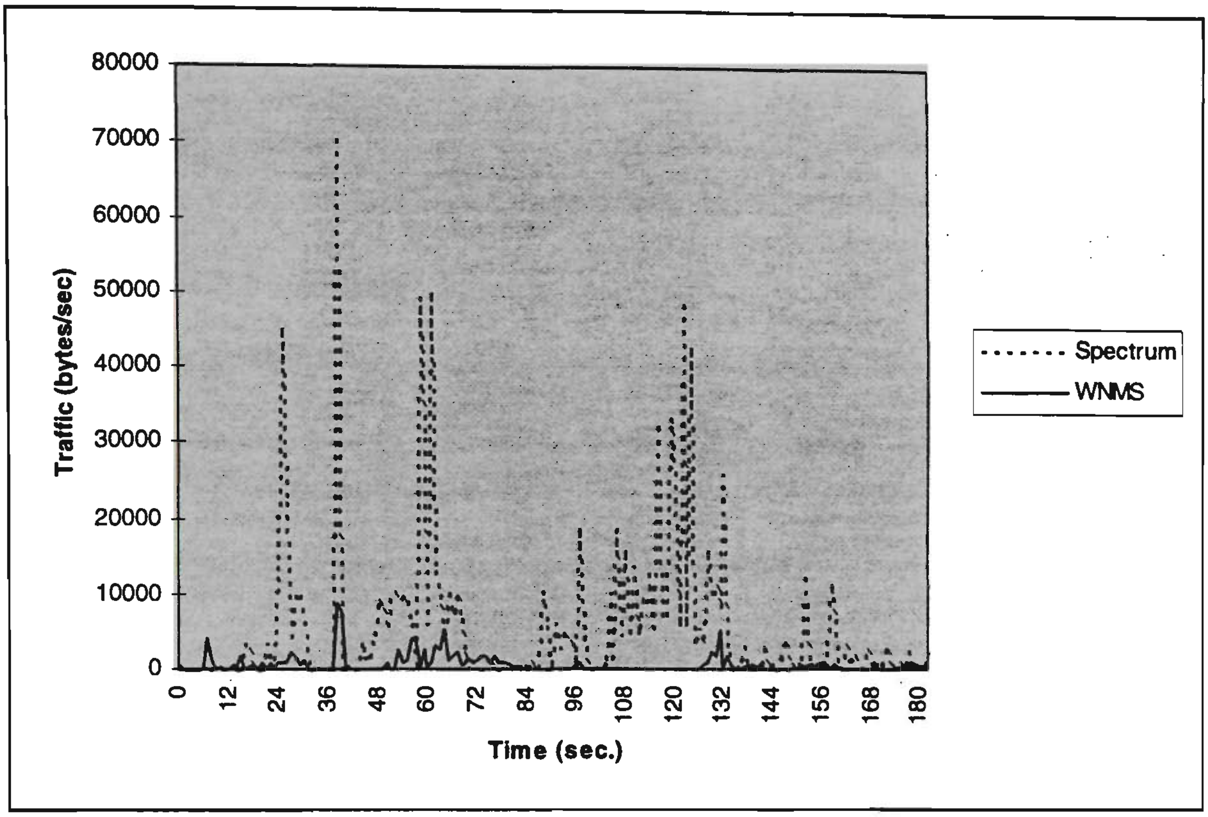

Figure 6.5- Comparing bandwidth requirements of WNMS vs. Spectrum.

It can be argued that collaboration data will increase, as the number of users increase. However, as shown in Table 4.1 of Chapter 4, the amount of data in a practical situation is quite manageable (2 Kbytes/sec). Besides, this is a feature not available in Spectrum or other commercial management systems (collaboration-aware in WNMS vs. collaboration-transparent in Spectrum).

6.5.2.7 Accessibility

Accessibility refers to the ability to easily access a system from many different locations. For network management systems, accessibility requirements are the availability of equipment and software and low bandwidth consumption. Bandwidth requirements were discussed in the previous subsection. Here, the availability of the equipment and software is discussed.

Most traditional network management systems are platform-dependent Unix-based programs, requiring a high-end graphical workstation. This not only restricts accessibility in terms of hardware requirements, but also requires installation of the software on all the machines, if they are not using the same file server. The latter problem causes additional system management complexity.

These problems do not exist for WNMS. Like most WWW-based systems, WNMS is platform independent. Moreover, as it is downloaded from the server on demand, it is not necessary to pre-install it (though pre-installation speeds up startup time). WNMS only requires installation of a general-purpose Web browser, which is currently available on most computers. This allows utilisation of virtually any computer found in the organisation's network for network management, provided adequate authentication and security are in place.

6.5.3 General Issues

As mentioned earlier, the general issues are issues that are not directly related to network management. In this section, the issues of workload, cost and concurrent operations are analysed and compared.

6.5.3.1 Workload analysis

NASA-TLX is a standard questionnaire developed by NASA Ames Research. It is a multi-dimensional rating procedure that provides an overall workload score (WWL) based on a weighted average of ratings on six subscales: Mental Demands (MD), Physical Demands (PD), Temporal Demands (TD), Own Performance (OP), Effort (EF), and Frustration (FR). The original questionnaire consists of a set of DOS-based programs. Initial testing revealed that the subjects did not like the interface to the questionnaire itself. Consequently, it was replaced by a Web-based hypertext questionnaire, as shown in Figure 6.6.

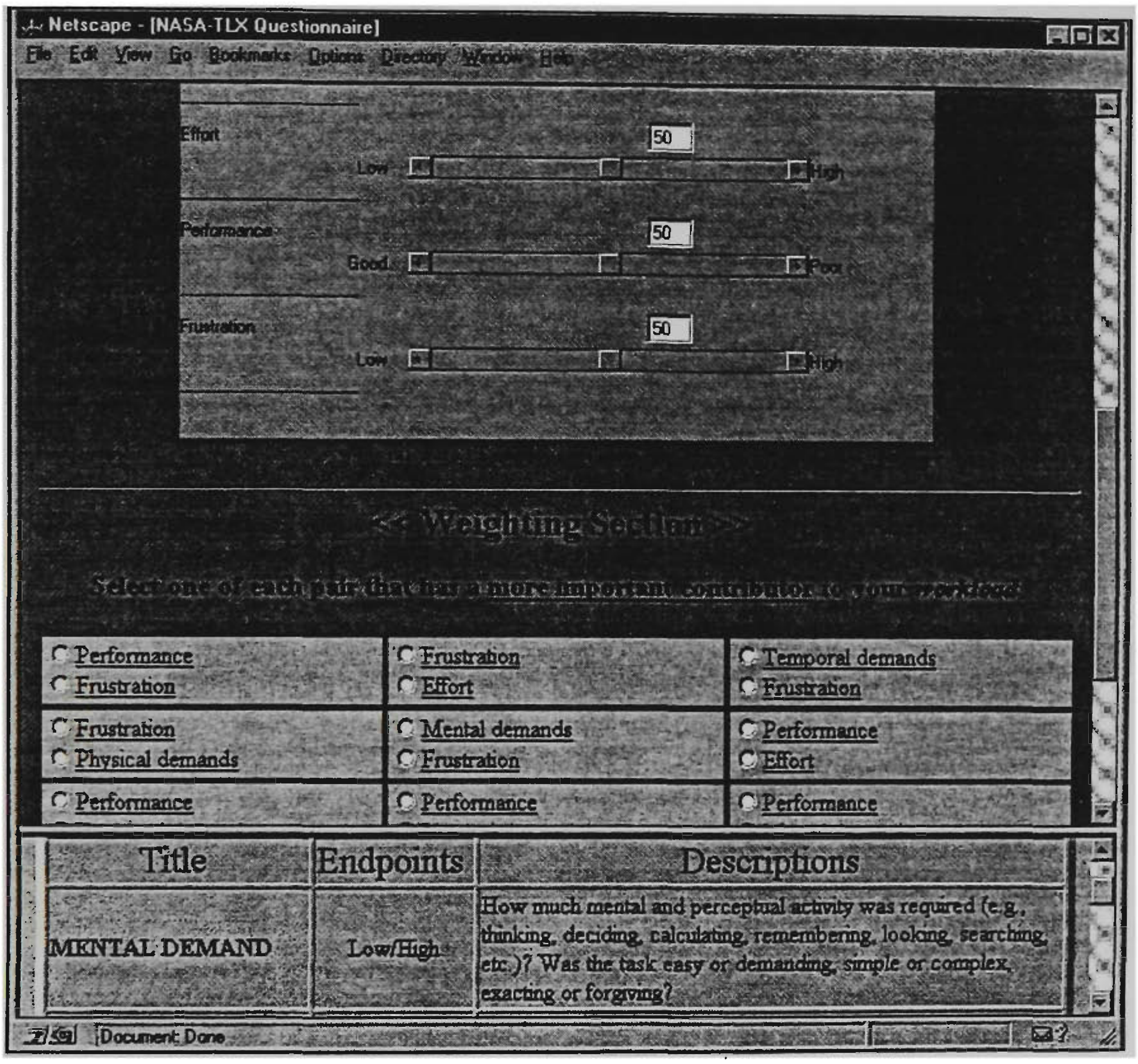

Figure 6.6- Web-based NASA-TLX questionnaire

The subjects were asked to complete the questionnaire after evaluating each user interface. They initially rated six items of the questionnaire, and then weighted those items against each other. Using the user's selection the overall WorkLoad is calculated and the form is sent to the server to be processed by a CGI script and stored in a file.

After all the subjects had completed the questionnaire, the average value for each factor is calculated (Table 6.4). In addition, the results of the *t-test* statistic are shown in the table to demonstrate if the observed differences are statistically significant or not.

	WWL	MD	PD	TD	EF	OP	FR
WNMS	25	21	32	25	27	16	18
Spectrum	56	55	49	49	61	47	54
t-test prob.	$p < 0.005$	$p < 0.005$	$p > 0.05$	$p < 0.005$	$p < 0.005$	$p < 0.0005$	$p < 0.001$
WNMS *better than* Spectrum	✓	✓	✗	✓	✓	✓	✓

Table 6.4- Comparison of workload indices of both systems

As the table shows, for all items WNMS has lower workload indices. In fact, in all categories, except physical demands (PD), the confidence level is quite high (a value of p<0.005 means that with the probability of 99.5% WNMS has a lower index). The result is exceptionally good for performance (OP) and frustration (FR), showing the satisfaction of the subjects with WNMS.

As shown in Table 6.4, the load index for physical demand (PD) of WNMS is not significantly better than Spectrum. This was expected as more physical activities are required to navigate and interact in 3D environments, especially when using a mouse. However, the absolute value of the index (32) is not too high to affect other indices of the overall workload. As better interaction devices emerge (eg. Phantom) it is likely that WNMS will outperform Spectrum-like WIMP interfaces in physical demands as well.

6.5.3.2 Cost analysis

In this section, a brief comparison of cost of WNMS and traditional user interfaces for network management is presented. It should be noted that WNMS only provides a user interface, and the cost of user interface is a fraction of the total network management system cost. However, the capability of WNMS to work in pure Web-based systems should be considered as well.

A traditional network management system not only costs more in terms of software, the cost of hardware equipment is higher too. A high-resolution graphical workstation usually cost several times more than a personal computer. This fact has been reinforced by the recent dramatic drop in price and increase in graphical power of PCs, so that nowadays, a high-level graphical PC is available on most desks. The platform independence feature of WNMS also allows utilisation of previously hardware investment, reducing the cost of buying new equipment.

As discussed in Section 6.5.3.3, the quick learnability feature of WNMS further reduces the cost of network management, by decreasing the cost of operator training.

WNMS can create more savings if it is used in a fully Web-based environment. Jander evaluated the costs of Web-based approaches with that of traditional systems, and concluded that the cost of a complete WWW based network management system is less than half of the cost of a commercial network management systems [Jander, 1996].

6.5.3.3 Concurrent operations

One of the interesting points raised at post-experiment interviews was the manner in which concurrent operations are handled by both systems. Any change in Spectrum requires switching to *edit* mode. When in edit mode, the corresponding view is locked, so that no other users can edit it at the same time. In contrast, in WNMS, the locking is based on objects. Users can lock an object, solely, rather than locking the whole view.

All users preferred the object-locking method, as it permitted more concurrent operations. However, as mentioned by some users, in specific situations a user might need to lock a set of objects, so that nobody can interfere with the operation. Therefore, it is desirable for the interface to provide facilities that allow a user to quickly lock a specific set of objects. In a 2D user interface, users can quickly drag the mouse to perform this action. In a 3D user interface, however, this task requires more investigation.

6.6 Discussion

In the previous sections, several methods for comparing a traditional user interface with WNMS were presented together with the result. The result show that in most cases WNMS is significantly superior to Spectrum. In this section, some of these results are further analysed.

It was expected that subjects would complain about the difficulty of navigation in WNMS. This is because the mouse is not a good device for navigation in 3D environments. However, the subjects evaluated the navigation method of WNMS as good (74 ± 6), and higher than that of Spectrum ($p<0.005$). One reason for this result could be that in WNMS some of the important locations in the 3D environment were defined as *viewpoints*. A viewpoint in VRML terminology is a location to which a user can quickly jump. For instance, users could quickly jump to the 'top viewpoint' to view all objects within a virtual world, or they could quickly jump to the neighbourhood of a specific object. It seems that this feature has compensated for some of the deficiencies of navigating with a mouse in a 3D environment.

WNMS also keeps a list of visited virtual worlds so that users can quickly navigate back to them. Inspired by these features, users suggested that the user interface provide a '*BACK*' button so that they could quickly return to the world they have just left. Note that the function of this button would be different to that of the browser, as it causes navigating back to the last virtual world, not undoing the last link jump. Furthermore, some subjects hinted that the provision of user-defined viewpoints would also be useful.

In contrast with navigation, some usability problems were found in the manner in which the interaction was performed. Within the interface, users can click on the name of the objects to get more information about them in the textual frame. Alternatively, they could click on the small sphere under the portal object to navigate into other views. To select and lock an object for manipulation, the object icon itself had to be clicked. However, compared with 2D interfaces, the subjects used to click on objects to select them to obtain more information about them or navigate into them. As this happened with almost all users, it appears to be a usability problem of the system, which requires more investigation. This might explain why there was no significant difference in terms of interaction between the systems ($p>0.05$).

In addition to the other issues discussed in previous sections, security issues are also important. Although other systems are prone to security problems, the situation is more critical with web-based systems as they are more accessible. Although the client-server architecture of WNMS addresses some of the security concerns, it is far from perfect. The provision of proper security is one important area for future work.

6.7 Conclusion

In this chapter, WNMS was evaluated and compared with a commercial network management system (Spectrum). Initially, some of the available method and techniques of evaluation were discussed, to show how the comparison could be performed.

As there were no guidelines to compare the systems against, a set of important factors was developed, based on a survey by Terplan, and interviews with network managers. After developing the guidelines, the systems were compared for each item in the guidelines.

To perform the comparison, an experiment was conducted, and some human subjects were asked to evaluate the system. Three different sets of subjects were used: novice, intermediate, and expert users, based on their prior experience with network management systems.

Subsequent tests and statistical analysis revealed that WNMS is superior to Spectrum in most cases. As Spectrum is representative of current commercially available network management systems, it is concluded that the techniques embodied in the WNMS interface are superior to current network management user interface techniques. While drawing this conclusion from a sample set of one may seem dubious, it can be supported by noting that current systems are almost identical in the way that they present information and in the way the user interacts with them.

In addition, it was discussed that although no significant difference could be found in interaction methods, it does not mean 3D user interfaces cannot provide better interaction facilities than 2D user interfaces. Rather, it was because of the usability problems of the interaction devices and methods used for WNMS. However, more investigation is required to address this problem.

Chapter 7

Conclusion and Further Research

7.1 Goals of the Thesis

In this chapter, the goals of this thesis are restated and some of the major findings of the thesis are reviewed. In addition, some areas for further research are presented. In general, this thesis investigates issues related to user interfacing of network management in forthcoming networks. The technology incorporated in these networks distinguishes them from traditional networks in several areas.

First, the abundant bandwidth and large size of these networks increases the complexity and importance of network management. In fact, the failure of some trunks would cost more than one million dollars per minute [Terplan, 1992]. This huge loss justifies utilisation of methods to minimise network failures, especially those occurring due to human errors. In case of failure, the quick detection and isolation of the fault is vital.

Other factors are the flexibility and remote configurability of the elements used in forthcoming networks. High flexibility allows better utilisation of network resources but also increases the complexity of network management. Consequently, the task of management should be distributed among a set of management stations to reduce the size and complexity of analysing data for each site.

The interdependence of private networks is another issue. The logical connectivity provided by these networks, permits several logically dependent networks to use the same physical layer. This causes a dependency among several networks, as the failure in the physical layer affects all of the involved parties. This fact urges co-operation among the network management systems of all involved parties in the management and monitoring of the shared resources.

These features are absent in current commercial network management systems. Indeed, the low semantic level of display and visualisation, and low level of interaction provided by two-dimensional user interfaces prevents them from satisfying the requirements of modern networks [Crucher, 1993]. The unsuitability of those interfaces, though not quantitatively proven, led to the investigation and development of several non-traditional user interfaces for network management as discussed in Chapter 2.

Psychological considerations are also important in user interfacing. Referring to Figure 1.2 of Chapter 1, there are two distances within the mental activity part of each user interface: semantic distance and articulatory distance. Semantic distance is the difference between the users' goal expression of the goal in the interface language, or the amount of processing required to determine if the goal is achieved. Articulatory distance represents the relation between the meaning of the expression and its physical format.

Shortening these distances, by providing a natural user interface or by directly mapping control variable to symbols in the user interface, provides more directness. The user interface can be designed so that relationship between intentions and actions and between actions and output seems straightforward and obvious. In such an environment, users feel engagement, first-personness and direct involvement with a world of objects rather than communicating with an intermediary [Hutchine, 1986].

The thesis was started with two major goals in mind: the design of a user interface that considers all of the factors mentioned above, and rigorous evaluation and comparison of the designed system with traditional ones. Chapters 3, 4 and 5 discuss the first goal, while the second goal is addressed in Chapter 6.

7.2 Key Results

Chapter 3 discussed the design and implementation of an immersive virtual reality system, utilising a combination of 3D input and output devices. By separating the network management part from the VR part, a modular system was implemented. The Cabeltron Spectrum network management system was used to provide the management facility. In addition, it was shown how object-oriented techniques could be used in the design of a user interface.

In Chapter 3, different methods for network hierarchy visualisation were also discussed. The discussion is summarised in Table 7.1. It was concluded that dividing the whole network into several worlds connected together via portals is a better approach due to its extensibility. Additionally, as shown in Chapter 5, a *cone tree model* can be used to visualise the whole network hierarchy.

	Method	Advantages	Drawbacks
1	Show all NEs within one view	View the whole hierarchy at one view	• Too many polygons, vey slow • Not extensible
2	Method 1, but using LOD to hide unnecessary details	View the whole hierarchy at one view	• File size too big • Not extensible
3	Separate worlds connected together via portals	• Extensible • fewer polygon, hence faster rendering	Cannot see the whole hierarchy
4	Method 3 in addition with simplified *cone tree* version of Method 1	Both advantages of Methods 1 and 3	—

Table 7.1- Comparison of different visualisation methods

The qualitative evaluation of the immersive system revealed the major problems of immersive VR user interfaces. The difficulty of presenting textual information, their inappropriateness for prolonged use and the prohibition of operator activities in the real world were the major pitfalls of the system. High levels of immersion, navigation and interaction were among the merits of the systems. In conclusion, it seems that the proper technology (higher resolution and lower weight of HMD, for example) is not yet available.

Inspired by the good visualisation facilities of 3D environments, it was decided to migrate towards a desktop VR system. The emergence of the VRML language and the rapid growth of the Internet and the World Wide Web during the development turned the system into a Web-based one. The high connectivity of the Internet and high availability of WWW browsers allowed the provision of collaboration among managers.

The architecture was also designed so that the facilities offered by modern networking technology could be utilised.

In order to support the multiuser features of the user interface, a protocol was defined to synchronise the view of the users. Later, noting the commonalities among multiuser interfaces, the protocol was generalised to become a Generic Protocol for Multiuser Interfaces (GPMI). The GPMI uses a client-server architecture, to provide consistency among users, with low bandwidth requirements. As this protocol has been designed to address the collaboration issues of multiuser interfaces, it provides the optimum support for those contexts. The protocol has been discussed in Chapter 4.

Incorporating the GPMI protocol and using WWW technologies, such as VRML, HTML, Java, JavaScript and CGI script, in conjunction with an object-oriented database, the second system, Web-based Network Management System (WNMS), was designed. WNMS was implemented as a proxy to one of the current network management systems, Spectrum. This method of implementation, allows the utilisation of the current infrastructure, while providing a coherent and consistent view of separate heterogeneous networks. It was also discussed that WNMS could be used in a pure WWW-based network management system.

WNMS was qualitatively evaluated and compared with the earlier immersive system. Although it proved to be a better approach than the immersive system, in general, it clearly lacks the high levels of navigation and interaction provided by that system. All issues related to WNMS were discussed in Chapter 5.

After initial empirical evaluation of WNMS, it was decided to use methods that are more rigorous to reveal the usability problem of the system, and compare it with a traditional user interface. This task was described in Chapter 6. User opinions, direct observation, workload analysis and heuristic evaluation methods were used to compare WNMS with Spectrum. A guideline was also developed to provide a basis for future evaluation and comparison of network management system user interfaces.

In general, WNMS proved to be a significantly better approach, and statistical analysis showed that in most cases, WNMS is clearly superior. For cases where there was no significant improvement, eg. interaction method and physical demands, the possible reasons were discussed, and the need for further investigation was shown. A summary of the result is presented in Table 7.2.

Criteria	WNMS better than Spectrum
Guiding fault management	✓
Integrated universal interface	✓
Learnability	✓
Centralised view	✓
Collaboration	✓
Bandwidth requirement	✓
Accessibility	✓
Overall workload	✓
Cost	✓
Security	✗

Table 7.2- The overal result of the comparison

In addition, a qualitative comparison of the immersive system, WNMS, and traditional network management user interfaces, showing the general strength and weakness of each approach, is given in Table 7.3. As the table shows, WNMS has some of the capabilities of both traditional and immersive systems. However, more investigation is required to remedy the interaction methods and security problems of this system.

System	Advantages	Drawbacks
Traditional 2D user interfaces	• Complete commercial versions currently available	• Low level of display • Low level of interaction • Expensive
Immersive VR user interface	• High levels of visualisation, navigation and interaction • Easy to learn	• Difficulty of presenting text • Technology not available • Expensive
WNMS	• High levels of visualisation and navigation • Easy to learn • Web-based systems are becoming popular	• Low level of interaction • Low Security

Table 7.3- Qualitative evaluation of different NMS user interfaces

WNMS ability to act as an interim model is another advantage as well. In fact, Web-based network management techniques are becoming so popular that some experts believe that they will replace the current systems [Bruins, 1996]. Therefore, WNMS can be instrumental in the transition period, as it allows network managers to utilise their existing management structure, while migrating to Web-based approaches. This provides a universal interface even in such an architecturally heterogeneous environment.

7.3 Further Research

A number of issues remained unresolved by this thesis. These could be topics for further investigation. These issues are listed in this section, with reference to the chapter where each issue was raised.

As concluded in Chapter 3, it appears that the proper technology is not yet available for a fully immersive VR user interface. However, in the near future, improvements in the quality of VR equipment and some modification to the architecture as discussed in Chapter 3, might allow utilisation of semi-immersive environments (eg. augmented reality) for telecommunication network management applications.

More investigations is also required into the adequate visualisation of network operation and performance. For instance, the status of the network can be mapped into a set of patterns, so that in case some parts of the network malfunction, the pattern changes. Using these techniques, human capabilities in pattern recognition are used to quickly filter the useful information from unwanted information.

The utilisation of sound, as both input and output devices, is another issue for further investigation. Using localised 3D sound can help to quickly reveal the whereabouts of an alarm. In addition, speech recognition techniques adds another natural input device, simplifying the navigation and interaction. However, more investigations is required to reveal how they can be employed, and if their application is worth the added complexity.

In Chapter 4, the GPMI protocol was developed. Although it has been defined to be general enough to support most systems, more research is required to prove its suitability for applications outside network management.

WNMS is one of the first systems to consider the use of a graphical 3D Web-based user interface for network management. Like other systems, the first design usually is not the optimum one, no matter how much effort has been used for its design. In addition, the rapid growth of Internet and Web-oriented programs provide more facilities that can be used to improve the system's performance. For example, as shown by other researchers [Hong, 1997],tools for real-time performance analysis of network operations are currently available. Integration of these facilities may improve the quality and performance of WNMS.

The security consideration in the Web-based systems is another potential area for future research. The high-level accessibility of these systems permits easier security breaches and justifies further investigation to satisfactorily resolve this problem.

Little research can be found in the literature evaluating 3D user interfaces or comparing them with 2D user interfaces. There are some reports on comparing usability of different 3D input devices [Grissom, 1989; Hinckley, 1997], though in primitive contexts. In general, the field of VR lacks the guidelines and standards available for their two-dimensional counterparts. The higher availability and affordability of VR systems urges the development of standards to guide system developers.

Similarly, the guidelines developed in Chapter 6 for network management systems comparison is far from perfect. This can be considered as the first attempt to provide a rigorous method for evaluation in this context. Further investigation is required to refine the guidelines, and provide methods for evaluating different aspects of network management systems.

This thesis has touched a wide-range of issues related to network management user interfaces, providing background for further research. This new direction can lead the evolution of traditional user interfaces, generating more effective user interfaces for the management of modern and high speed networks.

References

Armstrong, S., Freier, A., and Marzullo, K., "Multicast Transport Protocol", RFC 1301, Xerox, Apple, Cornell University, February 1992.

ANSI/IEEE Std 1278-1993, "Standard for Information Technology, Protocols for Distributed Interactive Simulation", March 1993.

Arlein, R., Gava, J., Gehani, N., Lieuwen, D., "Ode 4.2 (Ode<EOS>) User Manual", AT&T Bell Laboratories, April 1996.

Ary, D., Jacons, L., Razavieh, A., "Introduction to Research in Education", Fifth Edition, Harcout Brace College Publication, Orlando, USA, 1996.

ATM Forum, "Customer Network Management (CNM) for ATM public Network Service (M3 Specification) Rev. 1.05", January 1996.

Barrus, J., Waters, R., Anderson, D., "Locales: Supporting large Multiuser Virtual Environments", IEEE Computer Graphics and Application Magazine, November 1996, pp. 50-57.

Batini, C., Catarci, T., Costabile, M., Levialdi, S., "On Visual Representation for Database Query System", Informatique 93 Conference, France, March 1993.

Becker, R., Eick, S., Miller, E. and Wilks, A., "Dynamic graphics for Network Visualisation", in Proceedings of ISI, 1989, pp. 93-96.

Beadle, H.W.P., "Experiments in Multipoint Multimedia Telecommunications", IEEE Multimedia, 2(2), 1995, pp. 30-40.

Bell, G., Marrin, C., et al., "The VRML 2.0 Specification", 4 August 1996.

Bell, G., Parisi, A., Pesce, M., "The VRML 1.0 Specification", May 1995.

Benford, S. D., Bullock, A. N., Cook, N. L, Harvey, P. W., Ingram, R. J. and Lee, O., "A Spatial Model of Cooperation for Virtual Worlds", in Proc. Informatique '93 - Interfaces to Real and Virtual Worlds, Montpelier, France, March, 1993.

Berger, J., Dinh, L., Masiello, M., Scell, J., "NVR: A System for Networked Virtual Reality", International Conference on Multimedia Computing and Systems, 1994, pp. 37-44.

Biferno, M., "Mental Workload Measurement: Event-Related Potentials and Ratings of Workload and Fatigue", (NASA CR 177354). Moffett Field, CA: NASA Ames Research Center, 1985.

Briggs, P., "Usability Assessment for the Office: Methodological Choices and their implications", in Psychology issues of Human Computer Interaction in the Work-place, pp. 381-401, Elsevier, Netherlands, 1987.

Bruins, B., "Some experiences with emerging management technologies", The Simple Times, Vol. 4, No. 3, July 1996.

Card, S., Moran, T., Nowell, A., "The Psychology of Human Computer Interaction", Hillsdale, N.J., Erlbaum, 1983.

Carlson, c., Hagsand, O., "DIVE - a Multi-User Virtual Reality System", Proceeding IEEE VRAIS'93, 1993, pp. 394-400.

Chen, J., "WWW & VR Technologies for Network Management Applications", IEEE International Joint Symposia on Intelligence and Systems, November 1996, pp. 268-272.

Corbett, M., Macleod, M, Kelly, M., "Quantitative Usability Evaluation - The ESPRIT MUSiC project". in Human-Computer Interaction: Application and Case Studies, Amsterdam, Elsevier, pp. 313-38. 1993.

Crutcher, L; Lazar, A; Feiner, S; Zhou, M, "Management of Broadband Networks using a 3D Virtual World", Proc. 2nd Intl. Symp. on High Perform. Distributed. Computing, 1993, pp. 306-15.

Deri, L., "HTTP-based SNMP and CMIP Network Management", Internet Draft (draft-deri-http-mgmt-00.txt), December 1996.

Deri, L. and Manikis, D., "VRML: Adding 3D to Network Management", IS&N97, Italy, 1997.

Eick, S., Wills, G., "Navigating Large networks with Hierarchies", in Proceedings of IEEE Visualisation'92, 1993, pp. 204-210.

Eick, S., Fyock, D., "Visualizing Corporate Data", AT&T technical journal, January/Febryary 1996, pp. 74-86.

Eick, S., "Aspects of network Visualisation", IEEE Computer Graphics and Applications, march 1996, pp.69-72.

Ericksson, H., "MBONE: The Multicast Backbone", Communications of ACM, Aug 1994, pp. 54-60.

Farkouh, S. C., "Managing ATM-based Broadband Networks", IEEE Communications Magazine, May 1993, pp. 88-94.

Feiner, S., et al., "A Virtual World for Network Management", IEEE Intl. Symp. on Virtual Reality, 1993, pp. 55-61.

Fowler, H., "TMN-Based broadband ATM Network Management", IEEE Communication Magazine, March 1995, pp. 74-79.

Fuji, H., Nakal, S., Matoba, H., Takano, H., "Real-time Bifocal network Visualisation", IEEE, 1994, pp. 867-76.

Garland, E., Rowel, D., "Face-to-Face Collaboration", Byte magazine, Nov. 1994, pp. 233-242.

Greenhalgh, C., Benford, S., "Virtual reality Tele-conferencing: Implementation and Experience", Fourth European Conference on Computer Supported Cooperative Work (ECSCE'95), Sweden, Sep. 1995.

Grissom, S., "StEP(3D): A STandardized Evaluation Plan for Three-dimensional Interaction Techniques", International Journal of Human-Computer Studies, No. 43, 1995, pp. 15-41.

Grissom, S., Carlson, Perlman, "Tablet-based Valuators: A Comparison of Virtual Buttons and Hand Gestures for Interactive Computer Graphics", in Work with computers: Organizational, Management, Stress and Health Aspects, Elsevier Science Publishers B.V., Amsterdam, 1989.

Gunsthoevel, D., Boesser, T., "Predictive Metrics for Usability", In proceeding of the 4th International Conference on Human Computer Interaction, Stuttgart, September 1991.

Hagsand, O., "Interactive Multimedia VR in the DIVE system", IEEE Multimedia, Vol. 3, No. 1, Spring 1996, pp. 30-39.

Hagsand, O., Lea, R., "Using Spatial Technique to Decrease Message Passing in a Distributed VE system", VRML'97 Conference Proceedings, Monterey, CA, February 1997.

Harrison, B. and Mellquist, P., "Web Based System and Network Management", Internet draft (draft-mellquist-web-sys-01.txt), November 1996.

Hammond, N., Hinton, G., et al., "Evaluating the Interface of a Document Processor: A Comparison of Expert Judgement and User Observation", Proceedings of INTERACT'84, Vol. 2, pp. 135-9, Amsterdam, North-Holand.

Hill, S., Iavecchia, H., Byers, J., et. Al, "Comparison of four subjective workload rating scales", Human Factors. Vol. 34, No. 4, August 1992, p 429-439.

Hinckley, K., Tullio, J., Pausch, R., Proffitt, D., Kassel, N, "Usability Analysis of 3D Rotation Techniques", ACM CHI'97 Conference on Human factor in Computing Systems., 1997.

Hong, J., Kong, J., Yun, T., et al., "Web-Based Internet Services and Network Management", IEEE Communication Magazine, October 1997, pp. 100-110.

Houwing, E., Wiethoff, M., Arnold, A., "Usability evaluation from user' point of view: Three Complementary Measures". in Human-Computer Interaction: Application and Case Studies, Amsterdam, Elsevier, pp. 475-80. 1993.

Hutchins, E., Hollan, J., Norman, D., "Direct Manipulation Interfaces" In User Centred System Design: New Perspective on Human-Computer Interaction, Lawrence Erlbaum Associates, N. J., 1986, pp. 87-124.

ITU-T Recommendation M.3010, "Principle and Architecture for the TMN", Geneva, 1992.

Jander, M., "Web-based Management: Welcome to the Revolution", Data Communication, November 1996, pp.39-53.

Jeffries, R., Miller, J., Wharton, C., Uyeda, K., "User Interface Evaluation in the Real World: A Comparison of Four Techniques", proceedings of CHI'91 Conference, pp. 119-24, New York, ACM Press, 1991.

Kahani, M., Beadle, P., "Using Virtual Reality to Manage Broadband Telecommunication Networks", ATNAC'95 Conference Proceeding, Vol. 2, 1995, pp. 517-522

Kahani, M., Beadle, P., "WWW-based 3D Distributed, Collaborative Virtual Environment for Telecommunication Network Management", In ATNAC'96 Conference Proceeding, Australia, December 1996, Vol. 2, pp. 483-8.

Krueger, M., "Artificial reality II", Addison-Wesley, 1991.

Kositpaiboon, R. and Smith, B., "Customer network management for B-ISDN/ATM services", ICC'93 Geneva Conference proceeding, 1993, pp. 173-7.

Laurel, B., "Interfaces as Mimesis" In User Centred System Design: New Perspective on Human-Computer Interaction, Lawrence Erlbaum Associates, N. J., 1986, pp. 67-85.

Lauwers, J., Lantz, K., "Collaboration Awareness in Support of Collaboration Transparency: Requirement for the Next Generation of Shared-Window System" Proceedings ACM CHI'90 Conference, 1990, pp. 303-311.

Lazar, A; Choe, W; Fairchild, K; Hern, Ng "Exploiting Virtual Reality for Network Management", Communication on the move-ICCS/ISITA'92, 1992, Singapore, pp. 979-83.

Lea, r., Honda, Y., matsuda, K., "Virtual Society: Collaboration in 3D spaces on the Internet", http://www.csl.sony.co.jp/person/rodger/CSCW/cscw.html, 1996

Leigh, J., Johnson, A., Vasilikis, C., DeFanti, T., "Multi-Perspective Collaboration Design in Persistent Networked Virtual Environments", Proceedings of IEEE VRAIS'96, April 1996, pp. 253-260.

Leigh, J., Johnson, A., "Supporting Transcontinental Collaboration Work in Persistent Virtual Reality Environment", IEEE Computer Graphics and Application, July 1996, pp. 47-51.

Catarci, T., Costabile, M., Matera, M., "Visual Metaphors for Interacting with Databases", ACM-SIGCHI Boulletin, Vol. 27, N. 2, 1995, pp. 15-17

Levialdi, S., Massari, A., Saladini, L., "Visual Metaphor for Database Exploration", 1995.

Locke, J., Pratt, D., Zyda, M., "Integrating SIMNET with NPSNET Using a Mix of Silicon Graphics and Sun Workstations", 1992.

LW working group, "LW Specification Code and Comments", at http://www.livingworlds.com/draft_1/lw_code.htm, 24 February 1997.

Ma, M, Greening, D., Marvit, M., Brush, A., "An overview of Universal Avatars: Working Draft #1", at http://www.chaco.com/avatar/avatar.html, 15 August 1996

Macedonia, M., Zyda, M., et al., "Exploiting Reality with Multicast Groups: A Network Architecture for Large-scale Virtual Environments", Proceedings of the 1995 IEEE Virtual Reality Annual Symposium, March 1995, pp. 2-10.

Macedonia, M., Pratt, D., Zyda, M., "A Network Architecture for Large Scale Virtual Environment", Ph.D. Disertation, Naval Postgraduate School, Monterey, California, June 1995.

Macedonia, M., Zyda, M., "A Taxonomy of Networked Virtual Environment", IEEE Multimedia, Volume 4, No. 1, January-March 1997, pp. 48-56.

Macleod, M., Rengger R., " The Development of DRUM: A software Tool for Video-assisted Usability Evaluation", Proceeding of CHI '93 conference, Cambridge University Press, 1993.

Mandich, N., Belleli, T., "HCI Consideration in TMN systems", in *The management of Telecommunication Networks*, by Smith, R., et. al, Ellis horwood, 1992, pp. 251-260.

Marrin, C., "Proposal for a VRML 2.0 Informative Annex: External Authoring Interface Reference", Silicon Graphics, Inc., September , 1996.

Mastaglio, t. and Gallahan, R., " A Large-Scale Complex Virtual Environment for Team Training", IEEE Computer Magazine, July 1995, pp. 49-56.

Miller, D., Thorpe, J., "SIMNET, The Advent of Simulator Networking", Proceedings of the IEEE, Vol 83, No. 8, 1995, pp. 1114-1123.

Miller, J., Jeffries, R., "Usability Evaluation: Science of Trade-offs", IEEE Software, pp. 97-8, September 1992.

Monk, A., Wright, P., Haber, J., Davenport, L, "Improving Your Human-Computer Interface: A Practical Techniques", New York, Prentice-Hall, 1993.

Nielsen, J., "Finding Usability Problems Through Heuristic Evaluation", Proceedings of CHI'92 Conference, pp. 373-80, New York, ACM Press, 1992.

Nielsen, J., "Usability engineering at a Discount", In Designing and Using Human-Computer Interfaces and Knowledge Based Systems, pp. 394-401, Amsterdam, Elsvier, 1989.

Norman, D., "Cognitive Engineering" In User Centred System Design: New Perspective on Human-Computer Interaction, Lawrence Erlbaum Associates, N. J., 1986, pp. 31-61.

Pope, A.,"BBN Report NO. 7102, The SIMNET Network and protocols", Technical Report, BBN systems and Technologies, Cambridge, MA, July 1989.

Porteous, M., Kirakowski, J., Corbett, M., "Software Usability Measurement Inventory Handbook", Human Factors Research Group, University College, Cork, Ireland, 1993.

Pratt, D., Barham, P., et al., "Insertion of an Articulated Human into a Networked Virtual environment", Proceedings of the 1994 AI, Simulation and Planning in High Autonomy Systems Conference, December 94.

Preece, J., Rogers, Y., et al., "Human-Computer Interaction", Addison wesley, 1995.

Rauterberg, M., "Quantitative Measures for Evaluating Human-Computer Interfaces". in Human-Computer Interaction: Application and Case Studies, Amsterdam, Elsevier, pp. 412-17. 1993.

Rea, P., Whalley, S., "Advanced Interface into Network Management Workstation", IEE Colloquium on 'Distributed Virtual Reality', 1993, pp. 7/1-3.

Ribarsky, W., Bolter, J., Bosch, A. and Teylingen, R., "Visualization and Analysis Using Virtual Reality", IEEE Computer Graphics and Applications, January 1994., pp.10-12.

Robertson, G., Mackinlay, J. and Card, S., "Cone Trees: Animated 3D Visualizations of Hierarchical Information", Proceedings of the ACM SIGCHI '91 Conference on Human Factors in Computing Systems, New Orleans, LA, USA, April 1991, pp. 189-194.

Roehl, B., "Distributed Virtual Reality: An Overview", The Proceedings of VRML'95 Conference, at http://sunee.uwaterloo.ca/~broehl/distrib.html, 1995.

Roehl, B., "Some Thoughts on Behavior in VR Systems", The Proceedings of VRML'95 Conference, at http://sunee.uwaterloo.ca/~broehl/behav.html, 1995.

Roehl, B., "Channeling the Data Flood", IEEE Spectrum, Vol., 34, No. 3, March 1997, pp. 32-38.

Rumbaugh, J., Blaha, M., Premerlani, W., Eddy, F. and Lorensen, W., "Object-Oriented Modeling and Design", Englewood Cliffs, NJ: Prentice-Hall, 1991.

Sahin, V., "Telecommunications Management Network", in Aidarous, A. & Plevyak, T., "Telecommunications Network Management into the 21st Century", IEEE Press, 1994, pp. 72-121.

Sarkar, M., Brown, M., "Graphical Fisheye Views of Graphs", In Proceedings of ACM SIGCHI'92 Conference, 1992, pp. 83-91.

Shneiderman, B., "Direct Manipulation: A Step Beyond Programming languages", IEEE Computer Magazine, 1983, Vol. 16, No. 8, pp. 57-69.

Sidlor, D, "Managing Telecommunication Networks using TMN Interface Standards", IEEE Communications Magazine, Mar 1995, pp. 54-66.

Singh, G., "A Software Toolkit for Network-Based Virtual Environment", Presence, Vol. 3, No. 1, 1994.

Snowdon, D., West, A., Howard, T., "Towards the next generation of human-Computer Interface", in Informatiue'93: Interface to Real & Virtual Worlds, France, Mar. 93, pp. 399-408.

Snowdon, D. West, A., " The AVIARY VR-System. A Prototype Implementation", 6th ERCIM workshop, Sweden, Jun. 94.

Cabeltron Inc., "Spectrum Users' Manual V 3.0", 1995.

Srinivasan, R., "XDR: External Data Representation Standard", RFC1832, August 1995.

Stanger, J., "Telecommunication Applications of Virtual Reality", IEE Colloquium on Using Virtual World, pp. 6/1-3, 1992.

Stanger, J., "Networked Virtual Reality Applications", IEE Colloquium on 'Distributed Virtual Reality', 1993, pp. 1/1-1/4.

Stanney, K., Kennedy, R., "The Psychology of Cybersickness", Communication of the ACM, Vol. 40, No. 8, August 1997, pp. 66-68.

Sun Microsystem Inc., "Java Language Specification: version 1.0", 1996.

Sun Microsystem Inc., "Java Management API Architecture", June 1996.

Terplan, K., "Communications Network Management", 2nd ed., PTR Prentice Hall, New Jersey, 1992.

Thimbleby, H., "User Interface design", ACM Press, 1990.

Treu, S., "User Interface Evaluation: A Structured Approach", Plenum Press, USA, 1994.

Computer Associates Inc., "Unicenter TNG Product Description", at http://www.cai.com/products/unicent/tng.htm, 1997.

Walker, G., "Challenges in Information Visualisation", British Telecommunications Engineering Journal, Vol. 14, April 1995, pp. 17-25.

Walker, G., Rea, P., Whalley, S., Hinds, M., and Kings, N., "Visualisation of telecommunications network data", BT Technology Journal, Vol. 11, No. 4, October 1993, pp. 54-63.

WBEM Consortium, "Web-Based Enterprise management Proposal: Hypermedia Management Protocol Overview", Rev. 0.04, July 1996.

Welbank, M., "An Overview of Knwoledge Acquisation Methods" Interacting with Computers, Vol. 2, No. 1, 1990, pp. 83-91.

Wellens, C. and Auerbach, K. "Towards useful management", The Simple Times, Vol. 4, No. 3, July 1996.

Whalley, S., "Applications of Visualization within British Telecom", IEE Colloquium on Using Virtual World', 1993, pp. 6/1-3.

Wharton, C., Rieman, J., Lewis, C., Polson, P., "The Cognitive Walkthrough Method: A Practitioner's Guide", In Usability Inspection Methods, New York, John Wiley, 1994.

Whiteside, J., Bennett, J., Holtzblatt, K., "Usability Engineering: Our experience and evaluation", In Handbook of Human Computer Interaction, Amsterdam: North-Holland, 1988.

Wiethoff M., Arnold, A., Houwing, E., " The value of Psychological measures in Human-Computer Interaction", In proceeding of the 4th International Conference on Human Computer Interaction, Stuttgart, September 1991.

Appendix A

Details of Classes Used in the Immersive System

A.1 *InputDevice* class

This class is defined as an abstract class. All input device classes, such as keyboard and mouse, are subclasses of this class. Each input device can act as *head*, *body* or *pointer*. The job of input devices are to change the viewpoint or manipulate the objects. The viewpoint can be thought of as a window through which the user sees the virtual world. Some input devices, such as mouse, can accept any of the three jobs, while some, for example HMD, can only accept one job. The support for variety of input devices for each job allows the system to be operated using different configurations. That is, while it is possible to use the system using HMD, and glove, it is also possible to run it just using keyboard and a mouse.

The device that is assigned to act as a *head* controls the orientation of the viewpoint. For example if HMD (as an input device) is assigned to act as *head*, its rotation changes the orientation of the viewpoint. The position of the viewpoint is controlled by the input device that is acting as *body*. For instance, if a joystick is assigned to be the *body* device, by pushing its handle forward and backward, the view point moves forward or backward in the virtual world, correspondingly. The *pointer* device controls the pointer at the middle of the view enabling objects to be selected quickly. The use of *pointer* device is optional. In the absence of a pointer device, however, the body device should be used for object selection.

Because of the object-oriented approach, support for any new input devices can be easily added to the system by providing a device.

A.2 *OutputDevice* class

The *OutputDevice* class is quite simple, as most of its task is handled by the software library. The library provides support for the monitor (in VGA mode), and a few head-mounted displays. Whenever, a device other than the monitor is used, the output is shown on the monitor, as well. The *OutputDevice* class only has a pointer to the selected output device.

A.3 *World* class

The *World* class represents the virtual world in which network elements are represented and the user navigates. This class contains general information about the virtual world, and a list of pointers to all of the network elements in the world. The general information includes position and orientation of the viewpoint and the speed of movement or rotation within the world.

The reason for considering a different speed for each world is that the sizes of virtual worlds are different. For a big virtual world, a high movement speed is required, and for a small one, lower speeds are more appropriate. Although it is possible to easily adjust the speed, once it is done, it is saved so that next visit to the same world does not require the same adjustment.

A.4 *NetworkElement* class

NetworkElement class is an abstract class, which provides basic functionality to represent network elements. All specific network element classes are subclasses of this class. This class contains some information about the shape of each element in the virtual world, its position, and some other performance data, such as load and status. Note that some network elements, such as routers, appear in more than one virtual world. These objects are called portals and link worlds together. Portals have different positions in each world they are present in, so that changing their positions in one virtual world does not change their positions in others.

As this prototype interface was targeted for computer networks, rather than telecommunication ones, because of more accessibility, only elements used in those networks are defined. However, the object-oriented design of the system guarantees that other objects can be easily defined. The following network element subclasses defined for the prototype system. These subclasses are named after their counterparts in the real world.

- Router: Connects two virtual worlds together.
- MultiGate: Connects several virtual worlds together.
- LAN: Represent a segment in a Local Area Network.
- Generic SNMP: Any device that could not be correctly categorised by the NMS is considered "generic SNMP device". In the sample network, most bridges were categorised generic SNMP device by SPECTRUM.
- Pingable: Any simple network element that is represented only in one virtual world.
- Other: Any other network element not categorised above.

As the characteristics of each class is different from others, most functions within the *NetworkElement* class are defined as *virtual*, and each subclass defines the functions that it needs.

A.5 *Connection* class

This class defines the connection between network elements in a virtual world, reflecting the topology of the network. Objects that appear in several worlds, such as routers, have different connections in each world. So, each instance of *Connection* class should have a pointer to a *world*, as well.

Appendix B

Specification of VR Devices

B.1 CyberMan

The Logitech CyberMan is touted as a 3D controller that is ideally suited to the current rage in first-person perspective 3D environment games. It is built to circumvent the limitations of the mouse, giving six degrees of freedom to make movement natural and intuitive. Along with the ability to move the controller in the X, Y, and Z axes, the CyberMan also allows actions using pitch, yaw and roll manoeuvres. It also boasts tactile feedback, in the form of vibrations made by a small motor powered by two AA batteries or plugged into an AC adaptor. The CyberMan is shown in Figure B.1.

Figure B.1- The Logitech CyberMan.

The CyberMan is also able to coexist with a regular mouse attached to the computer. However, one would have to use the CyberMan mouse driver and include the command MOUSE DUAL in the autoexec.bat file. The regular mouse driver must be disabled.

Logitech CyberMan is currently discontinued, and a new version, called CyberMan 2, is introduced. The ergonomic issues has been improved in CyberMan 2 controller.

B.2 Data Glove

A 5th Dimension (5DT) Data Glove, shown in Figure B.2, was used for the project. This Glove consists of a small control box attached to a one-size-fits-all fabric glove. The control box is connected to the serial port of the computer via a cable. The 5DT Data Glove measures finger flexure and the orientation (roll and pitch) of a user's hand. It can also emulate a mouse as well as a baseless joystick and the user can also type while wearing the glove. These facts make it a suitable single input device for virtual reality applications. The Glove provides a natural interface, and is one of the necessary parts of immersive 3D environments.

Figure B.2- The 5DT Data Glove.

B.3 Head Mounted Display (HMD)

The Forte VFX1 HMD used for the project is a one-size-fits-all headgear, as shown in Figure B.3. The display consists of two full-colour LCD displays installed on a flip-up visor, which allows switching between monitor and the HMD without removing the HMD. Each LCD has a resolution of 789 x 230 pixel, and they can provide 3D stereoscopic imaging. The good features of the display are ability to adjust focusing and interpupillary distance (IPD).

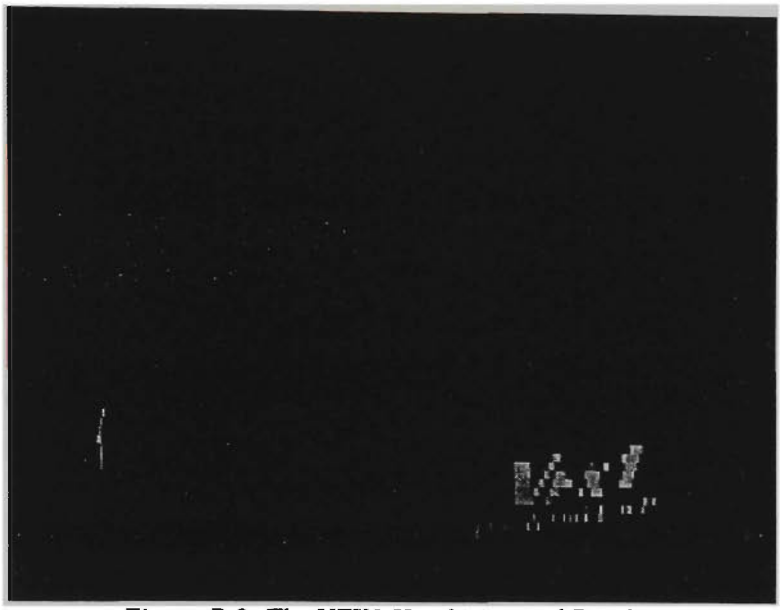

Figure B.3- The VFX1 Head Mounted Display.

The VFX1 HMD contains stereo headphones and a built-in microphone. In addition, it contains a tracker system that tracks the rotation of the head. The tracker has the vertical angular range of +/- 70 degrees and full 360 horizontal degrees angular range. It also detects yaw, pitch and roll rotation of the head.

The HMD is bundled with an small 2-axis device, called CyberPunk. The CyberPunk provides 3 programmable buttons and the ability to detect yaw and pitch rotation of the hand. Both the headgear and CyberPunk can act as a virtual mouse as well. The whole system can be connected to a PC via an interface card and a ribbon cable.

Appendix C

ASN.1 Code for GMPI PDUs

```
Generic_Protocol_Multiuser_Interface_PDUs DEFINITION ::=BEGIN


IMPORTS

    Unsigned Integer

        FROM RFC1832-XDR;


GPMIpdus ::= CHOICE {un-pdu    [1] IMPLICIT UNpdu,
                     undr-pdu  [2] IMPLICIT UNDRpdu,
                     o-pdu     [3] IMPLICIT Opdu,
                     lr-pdu    [4] IMPLICIT LRpdu,
                     m-pdu     [5] IMPLICIT Mpdu }

UNpdu ::= SEQUENCE { systemID       OCTET,
                     pduID          INTEGER {
                                         unpdu-val (1)
                                    },
                     pduType        OCTET,
                     pduSequence    Long,
                     timeStamp      Long,
                     piggyBack      Long,
                     senderID       ANY,
                     objectID       ANY,
                     attribType     OCTET,
                     attribID       ANY,
                     attribValue    ANY }

Long ::= Unsigned Integer
```

```
UNDRpdu ::= SEQUENCE { systemID    OCTET,
                       pduID       INTEGER {
                                       undrpdu-val (2)
                                   },
                       pduType     OCTET,
                       pduSequence Long,
                       timeStamp   Long,
                       piggyBack   Long,
                       senderID    ANY,
                       objectID    ANY,
                       attribType  OCTET,
                       attribID    ANY,
                       attribValue ANY,
                       speedVector ANY,
                       accelerationVector    ANY }

Opdu ::= SEQUENCE { systemID       OCTET,
                    pduID          INTEGER {
                                       opdu-val (3)
                                   },
                    pduType        OCTET,
                    pduSequence    Long,
                    piggyBack      Long,
                    action         OAction,
                    senderID       ANY,
                    objectType     ObjectType,
                    objectID       ANY,
                    objectShape    OCTETStringType }

OAction ::= CHOICE { create [1] IMPLICIT OCTET,
                     delete [2] IMPLICIT OCTET }

ObjectType ::= CHOICE { avatar [1] IMPLICIT OCTET,
                        non_avatar [2] IMPLICIT OCTET }
```

```
LRpdu ::= SEQUENCE { systemID      OCTET,
                     pduID         INTEGER{
                                        lrpdu-val (4)
                                   },
                     pduType       OCTET,
                     pduSequence   Long,
                     piggyBack     Long,
                     action        LRAction,
                     senderID      ANY,
                     objectID      ANY}

LRAction ::= CHOICE { lock [1] IMPLICIT OCTET,
                      release [2] IMPLICIT OCTET }

Mpdu ::= SEQUENCE { systemID       OCTET,
                    pduID          INTEGER {
                                        mpdu-val (5)
                                   },
                    pduType        OCTET,
                    pduSequence    Long,
                    timestamp      Long,
                    piggyBack      Long,
                    requestType    OCTET,
                    senderID       ANY,
                    recieverID     ANY,
                    message        ANY}

END
```

Appendix D

User Satisfaction Questionnaire

I. Questionnaire

Note: This survey was performed electronically. The questionnaire was presented in HTML format and the result was received via a CGI script. Therefore, the format has been slightly modified to suit printing.

1. General

1. Your email address, please?

2. Your occupation?

- O Academic Staff
- O Student
- O Designer/developer
- O Network manager
- O Others, Please specify:

3. Your Education?

- O PhD or MS
- O Bachelor
- O Undergrad student
- O No tertiary education

4. What is your main area of interest?

- O Network management
- O Virtual reality
- O User interface design
- O WWW-related issues
- O Others, Please specify:

5. What kind of computer do you use?

- ☐ PC
- ☐ Macintosh
- ☐ SUN workstation
- ☐ SGI workstation
- ☐ Others, Please specify:

6. How are you connected to the Internet?

- O Via 14.4K modem
- O Via 28.8 modem
- O Via LAN
- O Others, Please specify:

7. How much experience do you have in network management?

Very experienced O O O O O No experience

8. Which network management system you have ever used?

- ☐ Cabletron Spectrum
- ☐ HP OpenView
- ☐ IBM NetView
- ☐ Novel NMS
- ☐ Sun Net Manager
- ☐ Timeplex TIME/VIEW
- ☐ Others, Please specify:

9. Which network management system you use everyday?

- O None
- O Cabletron Spectrum
- O HP OpenView
- O IBM NetView
- O Novel NMS
- O Sun Net Manager
- O Timeplex TIME/VIEW
- O Others, Please specify:

10. How much experience do you have in virtual reality systems?

Very experienced O O O O O No experience

11. For how long you have been familiar with VRML?

- O More than 1 year
- O 6 - 12 months
- O 3-6 months
- O Less than 3 months
- O Haven't used it

12. Which VRML Browsers have ever used?

- ☐ SGI's CosmoPlayer
- ☐ Sony's Community Place
- ☐ Netscape's Live3D
- ☐ Dimension X's Liquid Reality
- ☐ Others, Please specify:

2. Navigation

1. How well do symbols in the virtual world represent their counterpart in the real world?

Very well O O O O O Very bad

2. How easy is moving around in the user interface?

Very easy O O O O O Very difficult

3. How fast does the system respond to the inputs?

Very well O O O O O Very bad

4. How well is the fault information (broken links, uncontactable devices) presented in the system?

Very well O O O O O Very bad

5. How well are the performance indicators (links' load, devices' colour) presented in the system?

Very well O O O O O Very bad

6. Comments on navigation issues

3. Interaction

1. How easy is the manipulation of the elements?

Very easy O O O O O Very difficult

2. Does the system provide enough feedback?

Quite enough O O O O O Not enough

3. Does the system respond to the inputs predictably and consistently?

Strongly Yes O O O O O Strongly No

4. How did you find the interaction with the system, overally?

Very well O O O O O Very bad

5. Do you think that utilisation of other input devices(eg. joysteak, DataGlove,..) would improve the interaction with the system?

Strongly Yes O O O O O Strongly NO

6. Comments on interaction issues

4. Collaboration

1. Did you used the collaboration feature of the system?

O Yes
O No - Go to the question 3

2. How did you find this feature?

Very useful O O O O O Not useful

3. In you opinion, is the textual collaboration enough for a network management system?

Quite enough O O O O O Not enough

4. Do you think that teleconferencing will increase the efficiency of the system?

Strongly yes O O O O O Strongly no

5. Comments on collaboration issues

5. Textual Data

1. Did you try the Textual frame of the system?

O Yes
O No - Go to the next section

2. How did you find it, overally?

Very useful O O O O O Not useful

3. Is the amount of textual information enough?

Quite enough O O O O O Not enough

4. How is the textual data presented in the user interface?

Very Structured O O O O O Unstructured

5. Comments on textual part

6. Learnability

1. Is the documentation adequate?

Quite enough O O O O O Inadequate

2. How long did it take you to master the user interface?

O Less than 10 mins
O 10-30 mins
O 30-60 mins
O More than 1 hour
O Didn't try

3. Did you have any problem finding the correct action?

Not at all O O O O O Always

4. Comments on learnability of the system

7. Overall Performance/Satisfaction

1. Is the system Consistent?

Very Consistent O O O O O Very inconsistent

2. How are the text and graphics parts mixed together?

Very well O O O O O Very bad

3. Do you believe that 3D graphics provides better visualisation than 2D graphics for network management?

Strongly yes O O O O O Strongly no

4. Are you satisfied with this system overally?

Completely satisfied O O O O O Completely unsatisfied

5. Would you recommend this system for everyday use?

Strongly recommend O O O O O Strongly reject

6. How do you see the future of Web based network management?(optional)

8. Comparison

1. With which network management system do you want to compare?

- O Cabletron Spectrum
- O HP OpenView
- O IBM NetView
- O Novel NMS
- O Sun Net Manager
- O Timeplex TIME/VIEW
- O Others, Please specify:

2. Using the WNMS, did you get a better visualisation of network hierarchy?

Strongly Yes O O O O O Strongly No

3. Using the WNMS, did you get better visualisation of network operation and performance?

Strongly Yes O O O O O Strongly No

4. Does the WNMS provide better navigation facilities?

Strongly Yes O O O O O Strongly No

5. Does the WNMS provide better interaction facilities?

Strongly Yes O O O O O Strongly No

6. Is WNMS easier to learn?

Strongly Yes O O O O O Strongly No

7. Using WNMS, are you less likely to make mistakes?

Strongly Yes O O O O O Strongly No

8. Do you prefer to use WNMS for network management?

Strongly Yes O O O O O Strongly No

9. Do you think WNMS is more suitable for the management of future high speed networks?

Strongly Yes O O O O O Strongly No

10. Room for general questions, comments and suggestions:

II. Results (raw data)

Questions	Q21	Q22	Q23	Q24	Q25	Q31	Q32	Q33	Q34	Q35	Q42	Q43	Q44	Q52	Q53	Q54
1st choice	1	2	2	6	7	7	5	4	4	4	3	6	3	4	2	2
2nd choice	6	13	7	7	5	8	9	11	10	2	2	7	6	11	8	8
3rd choice	8	0	4	2	3	0	1	0	1	9	10	2	5	0	5	5
4th choice	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0
5th choice	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Average	2.47	1.87	2.4	1.73	1.73	1.53	1.73	1.73	1.8	2.33	2.47	1.73	2.33	1.73	2.2	2.2

Questions	Q61	Q62	Q63	Q71	Q72	Q73	Q74	Q75	Q81	Q82	Q83	Q84	Q85	Q86	Q87	Q88	Q89
1st choice	6	4	4	6	9	6	3	3	15	10	6	5	8	8	5	6	5
2nd choice	9	11	9	8	6	9	11	9	0	5	7	10	3	7	3	9	9
3rd choice	0	0	2	0	0	0	1	3	0	0	2	0	4	0	7	0	1
4th choice	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
5th choice	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Average	1.6	1.73	1.87	1.73	1.4	1.6	1.87	2	1	1.33	1.73	1.67	1.73	1.47	2.13	1.6	1.73